HIGH-RESOLUTION GENE EXPRESSION ANALYSIS

by

Alyssa C. Frazee

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland

February, 2015

Abstract

RNA sequencing (RNA-seq) measures gene expression in cell populations at an unprecedented resolution. The advent of this new technology around 2008 spurred the need for new techniques for finding scientific meaning in the resulting data. Early statistical techniques for analyzing RNA-seq data were inspired by methods for microarray data analysis, and they involved quantifying gene expression by counting the RNA-seq reads falling within boundaries of pre-specified genes. However, RNA-seq data is very high-resolution, and much of that resolution is lost during the gene counting process. To that end, this thesis introduces novel statistical methods and software for analyzing RNA-seq data at a resolution beyond that of gene counting. First, we propose a technique for segmenting the genome into regions of differential expression between two population using single-base-level measures of signal. Next we focus on transcript-level differential expression analysis; in particular, we introduce tools for finding statistical differential expression signal in transcriptomes that were assembled *de novo* from RNA-seq reads. Finally, we create a tool for evaluating the statistical properties of RNA-seq differential expression methods: our new tool

ABSTRACT

generates RNA-seq reads to simulate an experiment with known transcript-level differential expression. These statistical and computational contributions to the RNA-seq analysis literature further our ability to draw meaningful biological conclusions from high-throughput RNA sequencing data.

Advisor: Jeffrey T. Leek, Ph.D.

Thesis Readers: Kasper D. Hansen, Anthony K. Leung, Steven L. Salzberg

Acknowledgments

To the faculty, staff, postdocs, and students in the Department of Biostatistics: thank you for all you've done for me. I couldn't have asked for a better or more supportive environment to do a PhD. Extra special thanks to my advisor, Jeff Leek. (See "Dedication.")

To my committee members - Kasper Hansen, Steven Salzberg, and Anthony Leung: thank you for being on both my final thesis committee and my preliminary oral exam committee. I really appreciate the valuable suggestions you've given me throughout my time here at Hopkins.

To the members of the genomics working group at Hopkins: thank you for letting me bounce ideas off of you, for helping me debug, for valuable research suggestions, for recommending that your collaborators use the software I developed, and for career advice.

To Ben Langmead, Kasper Hansen, Rafa Irizarry, John McGready, Marie Diener-West, and Karen Bandeen-Roche: thank you for being wonderful mentors, giving great advice, and giving me opportunities to improve as a researcher and teacher.

ACKNOWLEDGMENTS

To my friends in Baltimore: you made me feel at home even though I've been a thousand miles away from my family for almost five years. Thanks for bearing with me through the joys and challenges that grad school brings. To Melanie and Marie, my far-away friends: thank you for continuing to be truly supportive, awesome, and hilarious friends despite big life changes and awfully large distances between us.

To the Hacker School community: thank you for accepting me to the summer 2013 batch, for giving me the chance to dive deeply into programming and meet tons of truly amazing people in the process, for the opportunity to live in and fall in love with NYC, for fundamentally changing the way I viewed myself and my research, and for being unbelievably helpful during my job search.

Finally, to my family - my mom Shelley, my dad Dave, and my sister Kayla: Thank you for being you. Thanks for bearing with me during the terrible parts of this degree, and thanks for celebrating with me during the great parts. Thanks for making me laugh, making me feel like a superstar, and supporting me in every possible way. You're the best.

Dedication

There is only one person who even comes close to deserving this dedication, and that's Jeff Leek.

Thank you for believing in me, consistently supporting and respecting me, looking out for my well-being as a student and person, for being a fantastic model of truly enjoying your work, for inspiring me in both my research and my extracurricular statistics endeavors like blogging, for making meetings fun, and for bringing the lab together over baseball games and barbecues. Thank you for showing us that it's possible to be incredible at your job in addition to being friendly and humble. I so admire your fearlessness, confidence, resilience, and kindness.

You made this happen! Thank you so much.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

One of the fundamental principles of how living organisms function is outlined by the central dogma of molecular biology. The central dogma states, in essence, that cells function by transcribing pieces of their DNA sequence to RNA molecules, which exit the cell nucleus and are translated into necessary proteins.[1,2] The transcription step – transfer of information in a DNA sequence into an RNA molecule, called a *transcript* – is the beginning of the process called *gene expression*. Variation in gene expression plays a key role in several important scientific contexts: gene expression changes are involved in phenotypic differences between in human populations,[3] cells' cancer status,[4] and determination of a cell's identity.[5]

Scientists investigating questions in these and other areas related to gene expression have developed methods to measure a gene's expression level by quantifying the amount of RNA in the cell that was transcribed from that gene. One of the earliest

gene expression measurement techniques was the northern blot,[6] which size-separates a cell population's total RNA for easy quantification of specific sequences and radioactively or fluorescently labels the RNA corresponding to the gene sequence of interest. The northern blot is designed to measure expression of one gene at a time, which is useful when researchers have an idea of what they are looking for, but is less applicable to whole-genome studies. To that end, scientists developed other technologies to measure expression of many genes at a time, including expressed sequence tags[7] and quantitative polymerase chain reaction (qPCR).[8,9]

These techniques were good for measuring overall expression of genes. However, most genes emit not one but several different mRNA transcripts, called *isoforms*, through a process called *alternative splicing*,[10–14] and these older techniques for measuring gene expression did not give isoform-specific expression measurements. The DNA microarray revolutionized expression measurement by making it truly "high-throughput": microarrays make it very easy to get expression measurements for several thousand genes at a time.[15] They rely on knowing the sequences of the specific transcripts for which expression measurements are desired: known transcripts are assigned locations on the microarray chip, and isolated complementary DNA created from RNA in the cell is allowed to hybridize to those locations. Several effective statistical methods for normalization and differential expression analysis were developed for microarray data.[16–18] However, microarrays do not allow for easy quantification of alternative transcript expression, nor do they allow for discovery of expression out-

side of the probes being interrogated. These limitations are major drawbacks to the microarray: even in the organisms with the most well-characterized genomes, we have not yet annotated or discovered all the splice variants or the genomic regions capable of expression.

To that end, researchers have begun to rely on the current state-of-the-art gene expression measurement technique: RNA sequencing, or *RNA-seq*.[19] This technology directly sequences all of the messenger RNA (mRNA) present in a given cell population. RNA-seq works by first shearing isolated mRNA into short (200-600 nucleotide) fragments, and then reading the nucleotide sequence off of one or both ends of each fragment. Most popular RNA-seq protocols can only read short nucleotide sequences: about 100 bases at a time. However, if enough short reads are generated in an RNA-seq experiment, they are immensely useful: RNA-seq is potentially capable of measuring expression of novel splice variants or in regions not previously annotated,[20,21] and of measuring expression of all the isoforms of individual genes.[13,22] This flexibility, coupled with rapidly declining sequencing costs, has led to explosive growth in the use of RNA-seq technology in recent years.[23]

This thesis focuses on using RNA-seq data to detect *differential expression*, or the phenomenon in which a single genomic feature (exon, transcript, or gene) is expressed at different levels between two cell populations. The most popular existing differential expression analysis approaches fall into two categories, defined by how units of expression are defined and their expression quantified: (1) *annotate-then-identify*

and (2) *assemble-then-identify*. With microarrays, differential expression is relatively straightforward to quantify: intensity measurements from the same probe are compared across samples. In contrast, with RNA-seq reads, across-sample comparisons are not straightforward because the unit of expression measurement is not defined by the technology. Therefore, reads must be somehow summarized into units of expression before differential expression analysis can be performed. Annotate-then-identify methods do this summarization by counting the number of reads that fall within previously identified boundaries of known genes. On the other hand, assemble-then-identify methods seek to assemble full transcripts directly from the reads and perform transcript expression quantification based on likelihood methods. In both cases, differential expression analysis can then be performed on the resulting measurements at the gene or transcript level.

Here we make several contributions to the existing body of work on differential expression analysis of RNA-seq data. Chapters 2 and 3 propose a novel method for detecting genome-wide differential expression with RNA-seq that is more flexible than an annotate-then-identify approach while simultaneously avoiding the enormous challenge of full transcriptome assembly and quantification. In Section 2.1, we more fully describe the limitations with the two classes of existing approaches and propose our new intermediate class of methods, which we refer to as *identify-then-annotate*. We then propose and evaluate a specific implementation of the identify-then-annotate class of methods, called Differentially Expressed Region Finder (DER Finder). We

show that identify-then-annotate methods like DER Finder provide a good compromise between gene-counting methods and full transcriptome assembly methods.

In Chapters 4 and 5, we shift focus to the assemble-then-identify class of methods for detecting differential transcript expression. This chapter introduces and evaluates a new software package called *Ballgown*, which makes major inroads to solving some of the challenges facing users of assemble-then-identify approaches, particularly Cufflinks[22] and Cuffdiff.[24] Existing approaches suffered from a lack of flexibility in the types of experimental designs that could be analyzed, unreasonably conservative distributions of transcript-level p-values from differential expression tests, and extremely large computational time and memory requirements for experiments with more than a few biological replicates. Chapters 4 and 5 show these challenges can be solved by connecting transcriptome assemblies output by existing assemble-then-annotate methods to fast, flexible statistical models in R and the Bioconductor project.[25]

Finally, in Chapter 6, we introduce a new tool for simulating RNA-seq experiments with isoform-level differential expression. The tool, named *Polyester*, was designed to aid methods developers in testing their new approaches to isoform-level differential expression analysis, since the ground truth is almost never known in non-simualated RNA-seq data, and spike-in experiments are costly and difficult to do, especially at the transcript level. To our knowledge, no publicly-available, existing simulators have built-in mechanisms for specifying differential expression in transcripts at the read level, so *Polyester* was created to fill this need.

These contributions to software and methods for differential expression analysis using RNA-seq are steps toward a better understanding of the important research areas involving gene expression.

Chapter 2

Differential expression analysis of RNA-seq data at single-base resolution

This chapter describes work published in separate form in the journal *Biostatistics*, with contributions from co-authors Sarven Sabunciyan, Kasper D. Hansen, Rafael A. Irizarry, and Jeffrey T. Leek.

RNA-sequencing (RNA-seq) is a flexible technology for measuring genome-wide expression that is rapidly replacing microarrays as costs become comparable. Current differential expression analysis methods for RNA-seq data fall into two broad classes: (1) methods that quantify expression within the boundaries of genes previously published in databases and (2) methods that attempt to reconstruct full length RNA

transcripts. The first class cannot discover differential expression outside of previously known genes. While the second approach has discovery capabilities, statistical analysis of differential expression is complicated by the ambiguity and variability incurred while assembling transcripts and estimating their abundances. In this chapter, we propose compromise between these two classes: a novel method that first identifies differentially expressed regions (DERs) of interest by assessing differential expression at each base of the genome. Our method then segments the genome into regions comprised of bases showing similar differential expression signal, and then assigns a measure of statistical significance to each region. Optionally, DERs can be annotated using a reference database of genomic features. We compare our approach to leading competitors from both current classes of differential expression methods and highlight the strengths and weaknesses of each.

2.1 Introduction

In this section, we discuss the two classes of existing approaches to differential expression analysis with RNA-seq data and discuss how DER Finder's philosophy relates to these approaches.

RNA-sequencing generates millions or billions of short sequences from individual mRNA molecules. Analyzing these sequence reads requires several steps: First, each read must be matched to the position it originates from in the genome in a process

called alignment. Then, the number of reads aligned to specific regions must be summarized into quantitative measurements. The measurements are then normalized for the total number of reads measured for a particular sample and statistical models are applied to the summarized units. Oshlack, Robinson, and Young[26] describe this RNA-seq data analysis process in much more detail. Based on the summarization step, current statistical methods for the analysis of RNA-seq data can be grouped into two major classes. The methods in the first class, which we call *annotate-then-identify*, summarize the reads by counting the number that fall within pre-specified exons or genes. The exon and gene specifications, collectively called the annotation, are obtained from databases of previously identified genomic features.

Once the reads have been summarized at the exon or gene level, the statistical problem for the annotate-then-identify methods is similar to statistical analysis of microarray data, with some adjustments because the raw measurements are counts instead of intensities. The results from RNA-seq experiments can be naturally summarized into matrices like the results of microarray experiments, where rows are genes or exons and columns are samples. Therefore, many of the earliest statistical methods for analysis of RNA-seq data fall into the annotate-then-identify category because they were natural extensions of methods developed for microarrays. Two of the most widely-used annotate-then-identify methods are EdgeR[27,28] and DESeq;[29] Alexa-seq,[30] DEXSeq,[31] and a method developed by Wang et al.[32] are further examples of annotate-then-identify pipelines focusing on differential expression analysis of

genomic structures that may indicate splicing or transcriptional differences between groups.

The annotate-then-identify approach provides a straightforward and interpretable analysis, and tested statistical methodology is available once raw read counts have been summarized into a gene-level matrix. However, one disadvantage is that it relies heavily on the accuracy of annotation databases of gene and exon boundaries, and current annotation may be unreliable or hard to interpret.[33] As shown in Figure 2.1a, the annotated transcript structure at individual genomic loci can be complex. Biologically, the distinct but overlapping regions in vertical columns represent a single exon used slightly differently in multiple transcripts. This complexity requires the analyst to make important counting decisions in advance, since each distinct use of an exon (represented by a box in Figure 2.1b) represents a distinct potential counting region for annotate-then-identify methods. It is well known that different choices in how to count (all regions, only non-overlapping regions, or other choices) may lead to dramatically different results.[26,34] In the case shown in Figure 2.1, using a union model might allow for discovery of whole-gene differential expression, but it may mask a differential expression signal if, say, just one of the transcripts is overexpressed. Also, there is no "correct" gene model to use, so methods requiring this choice are at a disadvantage to those that do not. DER Finder does not require a gene model: if just a few transcripts or exons are differentially expressed, even in a complex scenario like Figure 2.1 shows, the gene will simply be flagged as displaying a complicated

Figure 2.1: Complex gene structures cause counting complications. (a) Structures of annotated transcripts in a 6kb region of the human genome (corresponding gene ID: ENSG00000099917). A transcript structure this complex causes problems in annotate-then-identify pipelines, as there is no clear way to determine which transcript or exon generated each read, especially if there is a high degree of overlap between unique features, as shown in panel (b): here, we zoom in on the exon on the right-hand side of panel (a) and see four overlapping yet distinct regions. Biologically, this could indicate a single exon with a varying transcription end site, but analytically, it introduces four potential counting regions and requires a critical counting decision to be made. Using a method like DER Finder eliminates the need for these decisions: if just one transcript or one form of an exon is differentially expressed, the genomic regions that uniquely identify that transcript or exon form will be called differentially expressed, and further analysis can be done on the small region to determine the exact phenomenon causing the observed pattern.

differential expression pattern. This type of result is not possible in a gene-model-based approach. A second disadvantage of annotate-then-identify methods is that they do not allow for discovery of novel or previously uncharacterized exons or genes, since they rely on previously-constructed databases.

The methods in the second class, which we call assemble-then-identify, attempt to assemble the full sequences of the mRNA molecules from which the short reads

originated. These methods rely less heavily on annotation databases of exon or gene boundaries. Another advantage is that assemble-then-identify methods aim to fully quantify all the potential isoforms of mRNA molecules emanating from each gene. However, the short length of typical sequencing reads leads to inevitable ambiguity when attempting to assemble and quantify abundances of individual mRNA molecules: it is virtually impossible to determine which of many possible sets of assembled transcripts truly generated the observed RNA-seq data. This ambiguity leads to varying and structured covariances between transcript measurements within genes, which complicates statistical analysis. There is also a high computational cost associated with assembling full transcripts, quantifying their abundances, and performing transcript-level statistical tests, as compared to the more direct annotate-then-identify approach. The most widely used algorithm in this category is Cufflinks/Cuffdiff;[22,24] others include Scripture[20] and IsoLasso.[35] In our experience, the computational cost of transcriptome assembly is non-trivial: for the 15-replicate experiment described later in this chapter (Section 2.3), Cufflinks took approximately 5 hours on 4 cores for each replicate, merging the 15 assemblies in preparation for DE analysis took 1 hour 39 minutes on 4 cores, and running Cuffdiff (assigning reads to transcripts and identifying DE) took about 42 hours on 4 cores, comparing expression in 9 of the replicates to that in the other 6. For comparison, alignment with TopHat took about 30 hours per sample on 4 standard cores. Other researchers[36] have confirmed that assembly with tools other than Cufflinks also took several hours,

and Cufflinks is one of the fastest assembly algorithms. Many of these tools allow the user to avoid the assembly problem by testing known transcripts for differential expression, but they then suffer from the previously mentioned shortcomings of annotate-then-identify methods.

Here we propose an intermediate class of methods which we call *identify-then-annotate*. These methods first summarize the RNA-seq experiment by counting the number of reads with alignments overlapping each individual base in the genome. Then they form a base-by-base statistic to identify nucleotides that are differentially expressed between groups. Consecutive bases showing a common differential expression signature are grouped into differentially expressed regions (DERs). The unit of statistical analysis is the DER, which can be evaluated for statistical significance using permutation or bootstrap approaches. DERs can then be compared to previous databases of exons and genes to identify: (1) regions of differential expression corresponding to known exons or genes and (2) novel regions of differential expression.

Currently, the closest analysis framework to an identify-then-annotate method is to combine pipelines: use an existing tool like rnaSeqMap[37] or an assembler like Cufflinks to identify expressed genomic regions, then test those regions for differential expression using existing statistical methods (e.g., DESeq[29] or EdgeR[27,28]). Another identify-then-annotate pipeline has been proposed in the form of MMD,[38] but it does not have a software implementation available and is designed to test known genes for differential transcript expression - not to be run on an entire genome. Here we propose

a new identify-then-annotate model that builds on the ideas behind the combining-pipelines approach: we feature a full statistical framework for expression detection and differential expression analysis.

The proposed identify-then-annotate model (1) allows for detection of differential expression in regions outside of known exons or genes, (2) allows for direct evaluation of differential expression of known genes and exons, (3) does not incur the added ambiguity and computational cost of assembly from short reads, and (4) can nonetheless detect differential splicing patterns and other expression differences between populations. Also, an identify-then-annotate tool can be used to address several commonly-posed research questions at once, including differential expression, splicing analysis, and detection of novel features. For example, we could analyze differential expression of known features with annotate-then-idenfity tools, then use an assembly tool to detect novel features, then re-run the annotate-then-identify tool to analyze differential expression of the novel features – but an identify-then-annotate tool would address all of these issues at once. The primary disadvantage is that the proposed class of methods does not allow for direct quantification of isoform-level expression. However, regions of potential alternative transcription can be easily identified where a subset of exons for a gene overlaps DERs but another subset does not, and those regions could be explored further with other tools.

2.2 Methods

2.2.1 Base-level Statistics

The first step in DER Finder is quantifying the evidence for differential expression at the nucleotide level. Since RNA-seq produces reads from mRNA transcripts, rather than directly from the genome, reads must be aligned using a strategy that accounts for reads that span intron-exon boundaries, called *junction reads*. In identify-then-annotate approaches like DER Finder, these junction reads are treated identically to reads that map directly to the genome when computing base-level coverage, but they should be aligned properly for correct quantification. TopHat[39] is an example of an aligner that appropriately handles junction reads. The user must make choices about mapping parameters to use during the alignment step: for example, some reads will map to more than one genomic location (due to, e.g., repetitive regions or pseudogenes). Non-unique read alignments can either be discarded, in which case repetitive regions would not appear to be expressed at all, or kept, which would allow all repetitive regions to appear expressed but would not allow those regions to be distinguished from each other. Whatever alignment strategy and corresponding parameters are used, the result is ultimately a large matrix with rows corresponding to bases and columns corresponding to samples; entries of this matrix are the number of aligned reads from a particular sample that overlap a particular nucleotide. We

refer to this matrix as the *coverage matrix*.

To quantify differential expression while accounting for biological variability and possible confounders, we fit a linear regression model to each row of the coverage matrix. Specifically, we let

$$g(Y_{ij}) = \alpha(l_j) + \sum_{p=2}^{P} \beta_p(l_j) X_{pi} + \sum_{k=1}^{K} \gamma_k(l_j) W_{ik} + \varepsilon_{ij} \tag{2.1}$$

where:

- Y_{ij} is coverage for sample i at location l_j
- g is a Box-Cox style transformation[40] (e.g., a log transformation) that makes the linear assumption acceptable
- $\alpha(l_j)$ represents the baseline gene expression (coverage) level at location l_j
- X_i is the covariate of interest for sample i (e.g., a 0/1 indicator variable for whether sample i is a case or a control)
- $\beta(l_j)$ is the parameter of interest that quantifies differential expression between cases and controls at location l_j (e.g., if g is a log transform, then $\beta(l_j)$ represents the log fold change in expression for cases compared to controls)
- W_{ik} $(k = 1, \ldots, K)$ are the values of K possible confounders for sample i, which may include sample-specific GC effect,[41,42] demographic variables like sex and age, or technical processing data. Including confounders in this model

is optional. We recommend setting W_{i1} to be some measurement of library size for sample i (e.g., median or 75th percentile of coverage for the sample across all bases)

- $\gamma_k(l_j)$ represents the effect of confounder k on gene expression at location l_j
- ε_{ij} represents residual measurement error at location l_j

The goal is to segment the genome into contiguous regions A where $\beta_p(l_j) \neq 0$ for at least one p for all $l_j \in A$. Instead of modeling $\beta_p(l_j)$ as a function (for example, with wavelet models or splines), we adopt a modular approach: we first estimate $\beta_p(l_j)$ for each location l_j, and then we divide the estimates into regions in a separate step. To estimate $\beta_p(l_j)$ along the genome and obtain test statistics from testing the null hypothesis that any of the $\beta_p(l_j) = 0$, we can use methods for estimating regularized linear contrasts,[16] which use a shrinkage approach that is appropriate for small sample sizes and borrows information across bases. Details of this approach are available in Section 3.1.1.

2.2.2 Identifying Candidate DERs with Segmentation

In this section, we refer to the test statistic resulting from the test for whether any $\beta_p(l_j) = 0$ as $s(l_j)$. (For ease of notation, we omit the j subscript in the discussion

that follows). For most experiments, we expect the function $s(l)$ to be a step function that is mostly 0, since most of the genome is not differentially expressed. We do not expect $s(l)$ to be smooth, because gene expression usually has a clear-cut start and end location. Hidden Markov Models (HMMs) are a natural way of modeling s, and we describe the specifics of our implementation here.

We assume there is an underlying Markov process along the genome, $D(l)$, with three hidden states: $D(l) = 0$ if $\alpha(l) = \beta(l) = 0$, $D(l) = 1$ if $\alpha(l) \neq 0$ and $\beta(l) = 0$, and $D(l) = 2$ if $\beta(l) \neq 0$. State $D(l) = 0$ corresponds to regions producing practically no gene expression. This state will be the most common, as most bases will not be covered by any reads because abundant gene expression is confined to a relatively small fraction of the genome. State $D(l) = 1$ corresponds to regions for which gene expression is observed but does not differ between populations. We are interested in finding regions in the differentially expressed state, $D(l) = 2$.

We assume that $D(l)$ is a first-order Markov chain with hidden state probabilities $\pi_d = \Pr(D(l) = d)$. We treat the transition matrix as fixed. By default, we set the retain state probabilities as very high with low transition probabilities between states, due to the sparsity of genes in the genome (Section 3.1.2; equation 3.1). The hidden state probabilities π_d can be roughly estimated based on the relative frequencies of bases covered or not covered by genes, along with a prior estimate of the number of differentially expressed genes. DER Finder results are largely robust to changes in the prior estimates for π_d (Section 3.2.3).

Conditional on the hidden state of each base l, we then assume $s(l)$ follows a normal distribution. Specifically, $s(l) \mid D(l) = d \sim N(\mu_d, \sigma_d^2)$. When $D(l) = 0$, there is little expression observed for base l, so we model the distribution as $N(0, \delta)$, where δ is an arbitrary, very small positive number. This distribution restrics $s(l)$ to values to very close to zero. We estimate π_0 empirically by calculating the fraction of bases where the average coverage is less than a threshold c. In our implementation, we considered a base to have hidden state $d = 0$ if none of the samples had coverage greater than 5.

The model parameters for states $D(l) = 1$ and $D(l) = 2$ (μ_1, μ_2, σ_1^2, and σ_2^2) can be estimated using a standard two-groups mixture model, first proposed for the analysis of differential expression in microarray experiments.[43] We assume that the statistics $s(l)$ from these two states are drawn from a mixture $f(s) = f_1(s)\pi_1^* + f_2(s)\pi_2^*$, where $\pi_1^* + \pi_2^* = 1$. (Estimates for π_1^* and π_2^* are scaled by the estimate of π_0 to obtain estimates for the overall state probabilities, π_1 and π_2, such that $\pi_0 + \pi_1 + \pi_2 = 1$.) Each mixture component is again assumed to be normal and can be estimated using the empirical null distribution defined in the two-groups model. We can then directly estimate the most likely path of unobserved states $D(l)$ based on the observed statistics $s(l)$ using standard estimation techniques for HMMs.

Further details on this approach to segmentation are available in the supplement to this chapter, including HMM parameter estimation (Section 3.1.2), specific form of the test statistics (Section 3.2.3), and validity of HMM assumptions (Section 3.2).

2.2.3 Statistical significance

The Hidden Markov Model essentially segments the genome into regions, where a region is defined as a set of contiguous bases having the same predicted hidden state. A region of bases with predicted hidden state $D(l) = 2$ is referred to as a *candidate DER*. After the segmentation step in the DER Finder pipeline, all analysis is done on the region level rather than the base level. Region-level analysis ensures the number of statistical tests is not unreasonably large, as it would be if we did a formal test at every base, and makes it such that variations in read coverage at individual bases that can arise due to technical artifacts in RNA-seq data will not affect the final results.

After segmenting the genome into regions, we assign a p-value to each candidate DER using a permutation procedure. In calculating the p-values for each candidate DER, we consider the size of the individual statistics within each region, since regions with very large test statistics are more likely to be truly differentially expressed. We apply an approach similar to Jaffe and others (2012):[44] first, we calculate the average base-level test statistic within each potential DER r: $\bar{s}_r = \frac{\sum_{l \in DER_r} s(l)}{\text{length}(DER_r)}$. Note that $\bar{s}_r$ is the region-level test statistic for region r. In the simple case-control scenario with no confounders, we can assign p-values to DERs with the following permutation procedure:

1. Permute the values of the covariate of interest (X_i) for all samples.

2. Re-calculate the base-level statistics using equation 2.1. Denote these null statis-

tics by $s^0(l)$.

3. Re-run the HMM on the $s^0(l)$s to identify a set of null DERs, indexed by ρ and denoted by DER^0_ρ.

4. To form region-level null test statistics, calculate the average base-level statistic within each null DER $\bar{s}^0_\rho = \frac{\sum_{l \in DER^0_\rho} s^0(l)}{\text{length}(DER^0_\rho)}$

Steps 1-4 are repeated B times, and the empirical p-value for region r is $p_r = \frac{1}{\sum_{b=1}^{B} P_b} \sum_{b=1}^{B} \sum_{\rho=1}^{P_b} \mathbb{1}(\bar{s}^0_\rho > \bar{s}_r)$, where P_b is the number of null DERs for permutation b. This quantity is the percent of null DERs with average statistic as or more extreme than the observed statistic for candidate DER r calculated on the observed data. Standard false discovery rate calculations can be applied to adjust these p-values for multiple testing. A statistical discussion of the validity of these p-value and FDR calculations is presented in Section 3.3.

In the case where confounders or additional covariates are included in model 2.1, a straightforward bootstrap extension of this permutation approach can be derived. After assigning statistical significance to each region, the DERs can be annotated using a reference database of known genomic features; an example of specific rules that could constitute an annotation procedure is described in Table 3.1.

2.3 Results: Comparison on Real Data

Our method is designed for differential expression detection in experiments with biological replicates, but many published experiments do not include such replicates.[45] We therefore designed an experiment comparing comparing brain tissue between 9 human males and 6 human females to assess the performance of competing methods: the Y chromosome was tested for differential expression between sexes using DER Finder, EdgeR and DESeq (using previously annotated exons) and Cufflinks/Cuffdiff. Specific details of the experiment are described in Section 3.4.

Two sets of results were obtained: one analysis compared males to females, and the other compared a randomly selected set of five of the males to the other four males. We expect virtually all genomic features of the Y chromosome (barring the pseudoautosomal region, pseudogenes, and other irregularities) to be differentially expressed between males and females, since females do not have a Y chromosome, and no genomic features to be differentially expressed between control males.

2.3.1 DER Finder results

DER Finder identified 534 Y-chromosome regions as differentially expressed ($q < 0.05$) between males and females. Six of these regions were classified as underexpressed in males, which we know to be artifacts since the whole Y chromosome is overexpressed in males, but the other 528 were identified as overexpressed in males as ex-

pected. Additionally, we found 280 novel differentially transcribed regions ($q < 0.05$). These novel transcribed regions ranged in length from 1 to 3814 bases, with only 19 of these regions having lengths less than 50 bases (Figure 2.2). These novel regions may indicate noise from the method (especially if they are very short), but they also may point to regions that should be examined further, either because they have interesting mapability characteristics or because they might truly be expressed and not yet annotated. The 534 differentially expressed regions pointed to 411 differentially expressed exons, using the criteria outlined in Table 3.1. These 411 exons came from 33 different genes, which means we found those 33 genes to be differentially expressed or indicate an event of interest. In comparing males to each other, we did not identify any differential expression on the Y chromosome: the minimum q-value for the regions found to be differentially expressed in the HMM step was 0.86.

2.3.2 Cufflinks/Cuffdiff results

Of 808 assembled transcripts tested for differential expression on the Y chromosme between males and females, the Cufflinks/Cuffdiff pipeline found no differentially expressed transcripts. The minimum q-value for these assembled transcripts was 0.45. While 736 of these transcripts showed nonzero abundance in males and zero abundance in females, these differences were not found to be statistically significant using the Cuffdiff methodology. Similar, too-conservative results were reported in the supplementary material of the manuscript accompanying the release of Cuffdiff

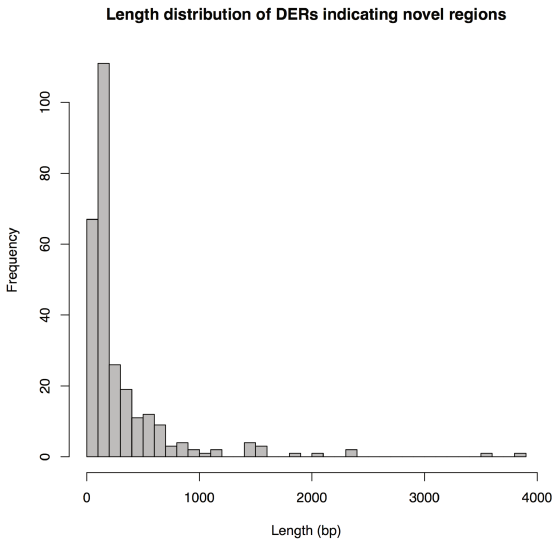

Figure 2.2: Distribution of lengths of DERs that indicate novel (un-annotated) differentially transcribed regions.

version 2.[24] In the comparison of normal males, none of the 818 assembled transcripts were called differentially expressed: the minimum q-value was 0.63.

2.3.3 EdgeR and DESeq results

Both of these methods tested 433 exons on the Y chromosome for differential expression between males and females. The other annotated exons on the Y chromosome did not have any reads mapping to them or the counting model did not allow for any reads to be counted for them. Of these 433 exons, EdgeR classified 113 and DESeq classified 115 as differentially expressed between males and females ($q < 0.05$). 97 exons were found by both EdgeR and DESeq. When comparing the males to each other, neither method found any exons to be differentially expressed: all except two

q-values (two exons that EdgeR assigned $q = 0.12$) were 1.

2.3.4 Comparison of results across methods

DER Finder exhibits performance comparable to that of EdgeR and DESeq, while all three methods outperform Cufflinks/Cuffdiff. DER Finder also has major advantages over EdgeR and DESeq: DER Finder is agnostic to annotation, which means it can identify differential expression signal in two important cases: (a) the case where a feature may be slightly mis-annotated or where the read mappings do not quite match up with the feature's annotation, and (b) the case where differential expression exists in regions that do not overlap annotated features. Both these scenarios occurred in the dataset studied. Figure 2.3a illustrates a case where the location or length of an exon may be incorrectly annotated. In that example, the mis-annotation caused the exon's expression to be underestimated when counting reads overlapping it. As a result, the statistical tests used in EdgeR and DESeq did not have enough power to call this Y-chromosome exon differentially expressed (both tools report $q = 1$) between males and females. DER finder more accurately reported the shown differentially expressed region as overlapping 61.3% of an annotated exon with $q = 0.001$. Also, DER Finder can find regions of interest that fall outside of annotated exons (Figure 2.3b). Closer inspection of the illustrated region reveals no Ensembl-annotated genes in the region, but shows that the expression is supported by five ESTs, providing evidence that the signal here is truly biological rather than simply background noise.

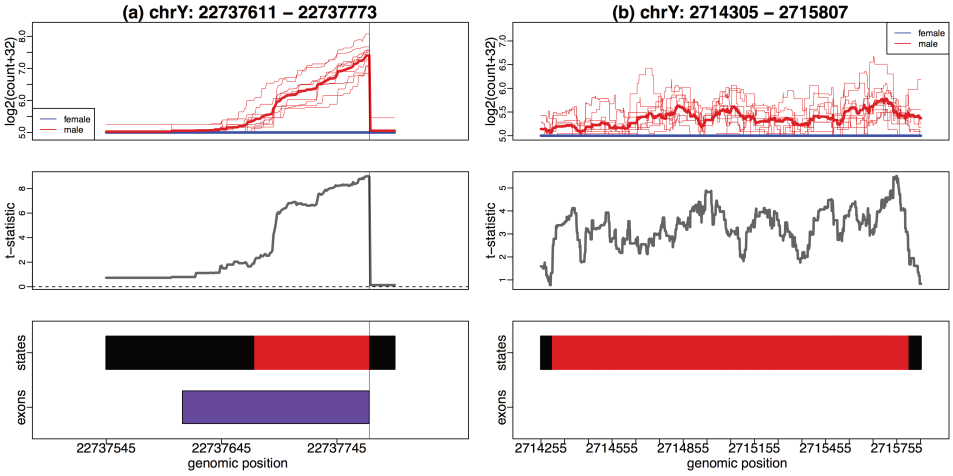

Figure 2.3: Cases where DER Finder correctly calls differential expression and annotate-then-identify methods do not. (a) Example of an exon (from gene *EIF1AY*, Ensembl exon id ENSE00001435537) whose location appears to be mis-annotated, leading EdgeR and DESeq to underestimate the exon's abundance and therefore incorrectly call this exon not differentially expressed. (b) Example of a differentially expressed region ($q = 0.001$) falling outside of an annotated exon, which can be found by DER Finder but not by EdgeR or DESeq. Though there are no annotated exons in this region, we believe this finding is more than noise because it is supported by the following annotated ESTs: CT001420, BF810102, BF369919, BF858017, CV424981 (GenBank accession numbers). Top panels: single-base resolution coverage (on log2 scale). Middle panels: t-statistics from linear model fit by DER Finder. Bottom panels: exon locations (denoted by purple boxes) and state calls from DER Finder: gray = not expressed, black = equally expressed, red = overexpressed in men.

Section 3.5 discusses other advantages of DER Finder over identify-then-annotate methods. Additional analysis of the agreement between DER Finder's results and those of Cufflinks/Cuffdiff, EdgeR, and DESeq can also be found in the supplementary chapter (Section 3.6). DER Finder's ability to find differential expression signal even in the presence of wrong or missing annotation is a key advantage over annotate-then-identify methods.

To assess whether the tools give reasonable results and to compare overall perfor-

mance of the three methods, MA plots[18] were used to show the relationship between each genomic unit's average expression (denoted with M) and the magnitude of differential expression it exhibits (denoted with A). The unit is a region for DER Finder, an exon for EdgeR and DESeq, and a transcript for Cufflinks/Cuffdiff. The MA plots resulting from the Y-chromosome experiment (Figure 2.4) reveal that DER Finder, EdgeR, and DESeq all produce reasonable results, but the findings from Cufflinks are somewhat problematic. While there does seem to be more overexpression of transcripts in males in the male/female differential expression analysis done by Cufflinks, we observe several extreme fold changes in the opposite direction, and the male-to-male comparison also produced these extreme fold changes. These problems do not exist in the other methods, whose MA plots illustrate high fold changes found between males and females and very little change found between males, as expected.

We note that in Figure 2.4, the displayed M and A values for EdgeR and DESeq are normalized. Specifically, the EdgeR plot is `logCPM` vs. `logFC`, where `logCPM` is $\log_2$ counts-per-million and `logFC` is the $\log_2$ fold change (male to female); both are normalized for library size and dispersion and are reported in the output of the `exactTest` function. The DESeq plot is $\frac{\log_2(\texttt{baseMeanA+0.5})+\log_2(\texttt{baseMeanB}+0.5)}{2}$ vs $\log_2(\texttt{baseMeanA+0.5})-\log_2(\texttt{baseMeanB}+0.5)$, where `baseMeanA` and `baseMeanB` represent library-size-normalized counts for males and females, respectively, and are reported in the output table from the function `nbinomTest`. Since `baseMeanA` and `baseMeanB` were sometimes 0, we added 0.5 as an offset to avoid calculating $\log_2(0)$.

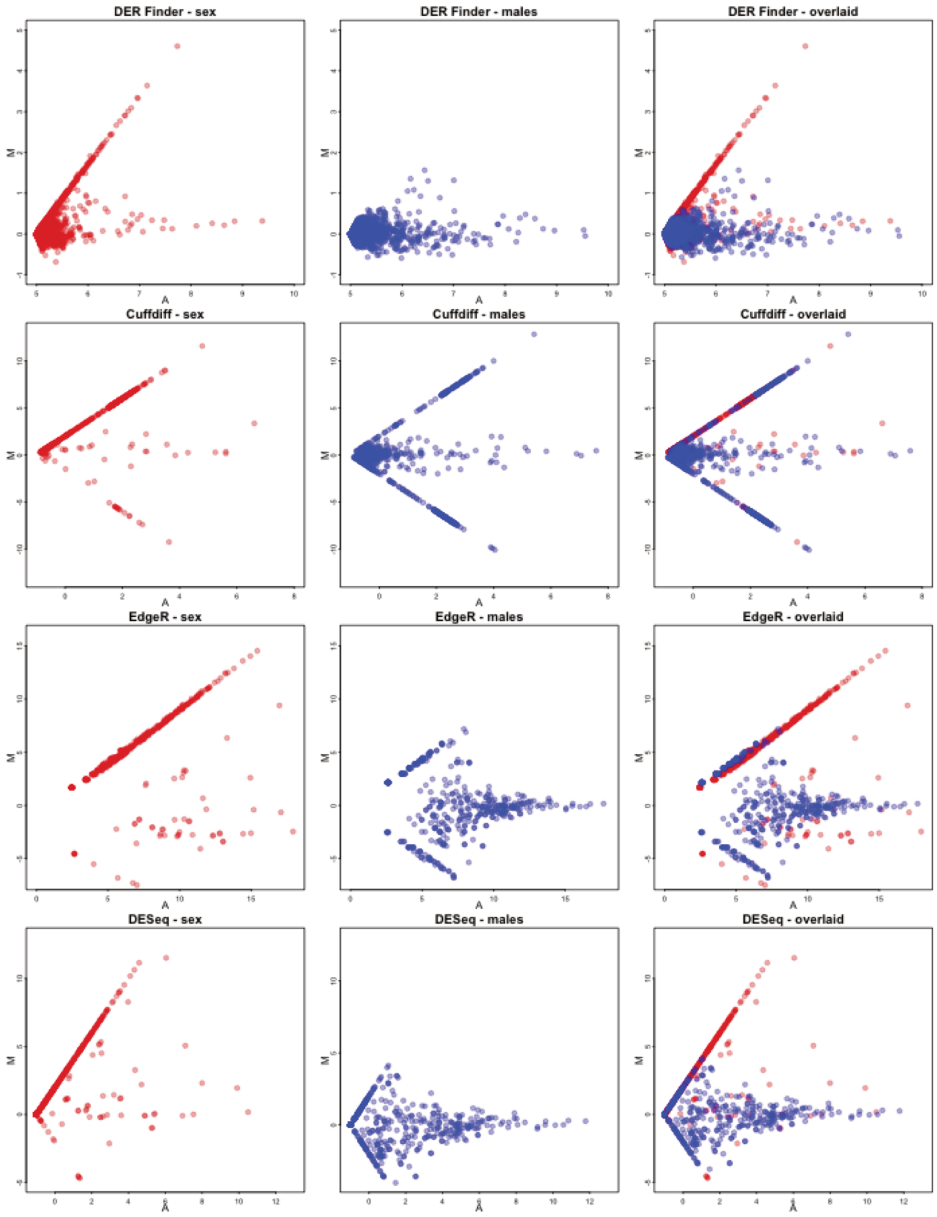

Figure 2.4: MA plots for Y chromosome regions, transcripts or exons, for each method and for both male vs. female (red) and male vs. male (blue) comparisons. On each plot, the x-axis represents the average log (base 2) abundance for each unit (region for DER Finder, transcript for Cufflinks, exon for EdgeR and DESeq), and the y-axis represents the log (base 2) fold change between males and females (red points) or the two groups of males (blue points). We expect to see the red, positively-sloped diagonal on all plots: this represents genomic regions expressed in males but not in females. In DER Finder, EdgeR, and DESeq, this diagonal corresponds with differential expression detected, however, no differential expression was detected in Cufflinks even though the red diagonal exists as expected.

Finally, to get a sense of each method's accuracy, we evaluated the tables of differentially expressed regions between sexes and between males produced by each method. We gathered all resulting regions – both negative results, from the male versus male comparison, and positive results, from the male versus female comparison – and ordered them by the value of their test statistic. An algorithm ranking all positive results ahead of the negative ones is preferred. Figure 2.5 shows, at each percentile of the differential expression test statistic, the percent of regions that are results from the male vs. female comparison. This is analogous to finding the percentage of findings that were truly positive at different significance cutoffs, assuming all tests in the sex comparison should be positives and tests in the male comparison should be negatives. We find that EdgeR, DESeq, and DER Finder perform comparably: all or most of the top 20% of regions, ranked by test statistic, came from comparisons between sexes. Cufflinks/Cuffdiff does much worse: only about 60% of the top 20% of their top transcripts came from the male-to-female comparison. DER Finder performs just slightly better than EdgeR and DESeq in addition to having other advantages over these methods, as discussed earlier.

2.4 Discussion

We propose DER Finder as a specific implementation of a new class of methods for differential expression analysis of RNA-seq data. The new class deals with identified

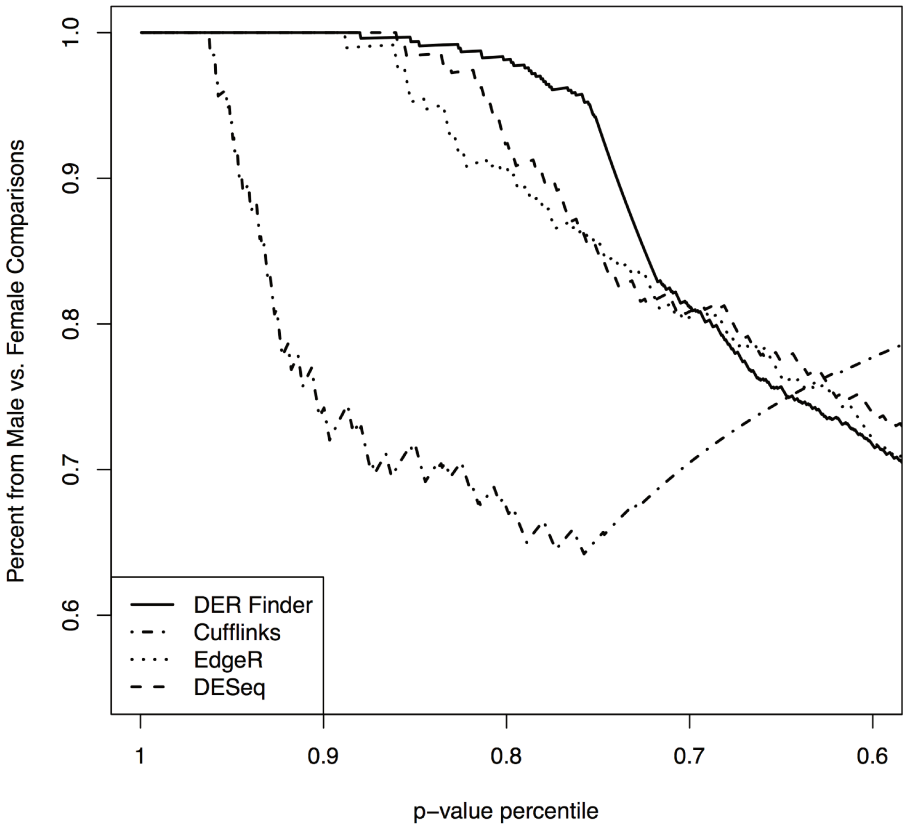

Figure 2.5: Percentage of significantly differentially expressed regions/transcripts/exons originating from male-to-female comparisons, using various percentiles of the p-value distribution as a significance cutoff. We find that most highly significant results are true positives, i.e., results with low p-values and high test statistics stem from comparing males to females, for DER Finder, EdgeR, and DESeq, while Cufflinks exhibits problems in this area.

challenges by (a) not relying on existing annotation when calling differential expression and (b) avoiding the immensely difficult problem of full transcript assembly by putting differential expression into a more straightforward framework. We have built on the ideas behind the approach of combining pipelines (e.g., rnaSeqMap combined with DESeq) to create a full pipeline for statistical analysis of differential expression. DER Finder outperforms Cufflinks/Cuffdiff and performs comparably to EdgeR and DESeq, while having the added advantages of sensitivity even in the presence of incorrect annotation and transcript discovery capability. We have also considered DER Finder's performance in other scenarios: a simulation study is presented in the supplementary material (Section 3.7.1) that addresses experimental design questions and examines DER Finder's accuracy. An identify-then-annotate method like DER Finder is an important step in developing new ways to analyze RNA-seq data, so further properties of these types of methods are worth investigating.

2.5 Software

The software and code used to do the analyses in this chapter are available on GitHub (`https://github.com/alyssafrazee/derfinder`).

2.6 Acknowledgements

We acknowledge helpful discussions with Geo Pertea and Steven Salzberg. Post-mortem brain tissue was donated by The Stanley Medical Research Institute's brain collection. We also thank Ms. Ou Chen for her technical assistance with high throughput sequencing.

Chapter 3

Supplementary Material: Differential expression analysis of RNA-seq data at single-base resolution

This chapter does not stand on its own, but provides supplementary material for Chapter 2. This chapter was published as supplementary material to the published form of Chapter 2 in the journal *Biostatistics*, with contributions from co-authors Sarven Sabunciyan, Kasper D. Hansen, Rafael A. Irizarry, and Jeffrey T. Leek.

3.1 Details on segmenting the genome into regions showing differential expression signal

3.1.1 Base-level test statistics

In DER Finder, we fit linear models (as specified by equation 2.1) at each base in the genome. To do this, we use methods for estimating regularized linear contrasts as implemented in the *limma* Bioconductor package.[16,46] We use a customized version of the `lmFit` function, keeping the default parameters. For the two-group comparison presented in Section 2.3, the test statistic $s(l)$ is a moderated t-statistic. This statistic is similar to the ordinary t-statistic obtained from testing whether $\beta_2(l) = 0$, but the standard error estimate for $\beta_2(l)$ used it its calculation is shrunk toward a prior variance estimate. This framework allows for borrowing of information across bases, which makes the statistical results more reliable in experiments with small sample sizes.

To be more specific, we present some of the details from Smyth (2004)[16] here; further details can be found in that paper. For ease of notation, since we are in two-group case, we drop the "2" subscript from $\beta_2(l)$ in the following discussion. Following Smyth's framework,[16] we assume a distribution on the estimated differential

expression effect at base l:

$$\hat{\beta(l)} \mid \beta(l), \sigma_l^2 \sim N(\beta(l), v_l \sigma_l^2)$$

where σ_l^2 represents the residual variance and v_l represents the unscaled variance at base l. We also assume a distribution on the estimated residual variance for the model at base l, assuming d_l is the residual degrees of freedom for that model:

$$s_l^2 \mid \sigma_l^2 \sim \frac{\sigma_l^2}{d_l} \chi_{d_l}^2$$

Then, a prior with parameters s_0^2 and d_0 is assumed on σ_l^2:

$$\frac{1}{\sigma_l^2} \sim \frac{1}{d_0 s_0^2} \chi_{d_0}^2$$

The prior describes how variances are expected to vary across bases. A prior is also assumed on $\beta(l)$ when $\beta(l) \neq 0$:

$$\beta(l) \mid \sigma_l^2 \sim N(0, v_{0l} \sigma_l^2)$$

This prior describes the distribution of differential expression parameters (here, log fold-changes) for differentially expressed bases. Under these priors, the posterior mean of ${\sigma_l}^{-2}$ given s_l^2 is $\tilde{s}_l^{-2}$, where:

$$\tilde{s}_l^2 = \frac{d_0 s_0^2 + d_l s_l^2}{d_0 + d_l}$$

Our test statistic $s(l)$, here the moderated t-statistic at base l, is then defined by:

$$s(l) = \tilde{t}_l = \frac{\hat{\beta(l)}}{\tilde{s}_l \sqrt{v_l}}$$

This empirical Bayes approach, where the posterior variance is used in the t-statistic calulation instead of the sample variance, is implemented in the `eBayes` function in *limma*. Data-driven estimation of the values of d_0 and s_0^2 is built into the `eBayes` function, as described in Section 6 of Smyth (2004).[16]

3.1.2 Hidden Markov Model

Once the nucleotide-level test statistics have been calculated, a Hidden Markov Model is fit on those statistics. In the general case, described in Section 2.2.2, we assume a three-state Markov process D along genomic locations l, such that $D(l) = 0$ when base l is not expressed, $D(l) = 1$ when base l is equally expressed between conditions, and $D(l) = 2$ when base l is differentially expressed. However, in our implementation, we found it convenient to divide the differentialy expressed state into two separate states. So in DER Finder, we define $D(l) = 0$ and $D(l) = 1$ the same way we do in the general case, but we assume here that $D(l) = 2$ corresponds to overexpression of base l in cases (compared to controls) and $D(l) = 3$ corresponds to underexpression

of base l in cases.

As input, the HMM requires several parameters: a transition matrix (defining probabilities of transitioning from one hidden state to another in consecutive base-pairs), fixed marginal probabilities of being in each hidden state, and parameters defining the distribution of $s(l) \mid D(l)$. For transition probabilities, DER Finder uses the following matrix as the default:

$$\begin{bmatrix} 0.999 & (1/3) * 0.001 & (1/3) * 0.001 & (1/3) * 0.001 \\ 0.001 - 2 \times 10^{-12} & 0.999 & 1 \times 10^{-12} & 1 \times 10^{-12} \\ 0.001 - 2 \times 10^{-12} & 1 \times 10^{-12} & 0.999 & 1 \times 10^{-12} \\ 0.001 - 2 \times 10^{-12} & 1 \times 10^{-12} & 1 \times 10^{-12} & 0.999 \end{bmatrix} \tag{3.1}$$

Entry (k, k') $(k = 1, 2, 3, 4)$ of 3.1 defines $Pr(D(l) = k-1 \mid D(l-1) = k'-1)$. Low probabilities are intentionally assigned to transitions from a differentially expressed state to an equally expressed state and vice versa, based on the assumption that discrete genomic features are not usually only partially differentially expressed. These parameters may be changed by the user. Initial tests of DER Finder indicate the method is not sensitive to changes in the parameters of the transition matrix as long as the diagonal entries are reasonably large.

The parameters left to estimate are $\pi_d = Pr(D(l) = d)$, μ_d, and σ_d^2 for $d = 0, 1, 2, 3$. Recall that we assume $s(l) \mid D(l) = d \sim N(\mu_d, \sigma_d^2)$. We estimate $\hat{\pi_0}$ as the

fraction of bases where average coverage is less than some threshold c, as described in Section 2.2.2. In that scenario, we used a slightly modified rule where we estimated the fraction of bases where no replicates had coverage of at least 5. Estimates of $\hat{\pi_1}$, $\hat{\pi_2}$, and $\hat{\pi_3}$ are obtained from the maximum likelihood approach of the two-groups model.[43] This model estimates $\hat{\pi_1}$ directly, and we assume that $\pi_2 = \pi_3$, i.e., that differential expression in either direction is equally likely. Thus we estimate both $\hat{\pi_2}$ and $\hat{\pi_3}$ as $(1 - \hat{\pi_0} - \hat{\pi_1})/2$. The two-groups model also gives estimates for $\hat{\mu_1}$ and $\hat{\sigma_1^2}$, and we assign $\hat{\mu_0} = 0$ and $\hat{\sigma_0^2} = 1 \times 10^{-7}$, requiring virtually all emissions from state 0 to be 0. Finally, we estimate $\hat{\mu_2}$, $\hat{\sigma_2^2}$, $\hat{\mu_3}$ and $\hat{\sigma_3^2}$ with a data-driven method. We will describe the procedure for estimating $\hat{\mu_2}$ and $\hat{\sigma_2^2}$; the method for $\hat{\mu_3}$ and $\hat{\sigma_3^2}$ is similar.

Define n to be the total number of nonzero t-statistics that were generated from differential expression tests. The two-groups model is only run on these n t-statistics, which means that it gives a direct estimate of what we will call π_{0nz}, i.e., the percentage of nonzero t-statistics with true state $D(l) = 1$. Then $\hat{\pi_1}$ is estimated as $\pi_{0nz}(1 - \hat{\pi_0})$.

Next, define the function n.above(x) as the observed number of nonzero t-statistics greater than x. Also define the function $c(p) = \hat{\sigma}_1\Phi^{-1}(p) + \mu_1$, where Φ represents the cumulative distribution function of the standard normal distribution. Note that for $p \in [0, 1]$, $c(p)$ yields the $100p^{th}$ percentile of the normal distribution for the equally expressed t-statistics, i..e, t-statistics emitted from bases with hidden state $D(l) = 1$. Using an iterative procedure, and using our estimate for π_{0nz}, we find the

value $p \in [0, 1]$ such that

$$\text{n.above}[c(p)] - (1-p)\pi_{0nz}n = 0.25(1-\pi_{0nz})n \tag{3.2}$$

The reason behind finding this p is as follows: note that $0.25(1-\pi_{0nz})n$ is the estimate of half the number of nonzero t-statistics corresponding to bases with $D(l) = 2$: $(1-\pi_{0nz})n$ is the estimated number of differentially expressed bases ($D(l) = 2$ or 3), half of those have $D(l) = 2$, and we multiply by 0.5 again to get half that quantity. Also note that $(1-p)\pi_{0nz}n$ gives the expected number of equally expressed t-statistics ($D(l) = 1$) greater than $c(p)$. Thus, the difference between the number of observed t-statistics greater than $c(p)$ and $(1-p)\pi_{0nz}n$ should yield the number of t-statistics with $D(l) = 2$ that are greater than $c(p)$. When we find a p such that this difference equals half the estimated number of t-statistics with $D(l) = 2$, we can use $c(p)$ as an estimate for the median of the distribution of overexpressed t-statistics. Since we assume this distribution is normal, $c(p)$ also provides an estimate for its mean, μ_2.

We can use $\hat{\mu_2}$ to estimate $\hat{\sigma_2^2}$: assume that p solves 3.2 above, and choose any value p' in $(p, 1)$. Define the quantity:

$$q = 1 - \frac{\text{n.above}[c(p')] - (1-p')\pi_{0nz}n}{(1-\pi_{0nz})0.5n} \tag{3.3}$$

The numerator of the fraction in 3.3 gives the estimated number of overexpressed t-statistics greater than $c(p')$, and the denominator gives the estimated total number of

t-statistics with $D(l) = 2$. Therefore, q denotes what percentile of of the distribution of $s(l) \mid D(l) = 2$ is given by $c(p')$. Then, since we know $\Phi^{-1}(q)$, $c(p')$, and $\hat{\mu_2}$, we can solve the equation

$$\Phi^{-1}(q) = \frac{c(p') - \mu_2}{\sigma_2} \tag{3.4}$$

for the unknown σ_2, to get an estimate $\hat{\sigma_2}$. We then estimate $\hat{\mu_3}$ and $\hat{\sigma_3^2}$ analagously.

Numerical failure can occur in estimating $\hat{\mu_2}$, $\hat{\mu_3}$, $\hat{\sigma_2^2}$, and/or $\hat{\sigma_3^2}$. As backup, we estimate $\hat{\mu_2}$ with the 95th percentile of a normal distribution with mean $\hat{\mu_1}$ and variance $\hat{\sigma_1^2}$, $\hat{\mu_3}$ with the 5th percentile of that distribution, and $\hat{\sigma_2^2}$ and $\hat{\sigma_3^2}$ with whatever was estimated for $\hat{\sigma_1^2}$.

Simulation studies comparing our data-driven method to an EM algorithm, implemented with the *mclust* package,[47] suggest that our algorithm is more conservative (i.e., distributions for $s(l) \mid D(l) = 2$ and $s(l) \mid D(l) = 3$ are estimated to be further from the distribution of $s(l) \mid D(l) = 1$) and more computationally efficient than the EM algorithm.

Using all these pre-set and estimated parameters, the HMM is fit in DER Finder using a Viterbi algorithm.[48] For our implementation, we used the `dthmm` and `Viterbi` functions from the *HiddenMarkov* R package.[49] By default, a non-stationary, homogenous HMM is fit (non-stationarity is the default in `dthmm`), though the user may fit a stationary HMM if desired. The model outputs the most likely state for each base in the genome given the observed t-statistics.

Runtime. The analysis done in Chapter 2 (Y chromosome analysis) took about 1 hour to run. Larger chromosomes took longer with the software we used when Chapter 2 was written: statistical analyses of chromosomes 1 and 12 took about 27 hours and 8 hours, respectively. The current release of the `derfinder` software[50] is available from Bioconductor[25] and is much more efficient than the version used in this chapter.

3.2 HMM Assumptions

Here we provide explanations and empirical evidence regarding the assumptions made in the HMM step in the DER Finder pipeline.

3.2.1 Correlation

DER Finder by default fits a first-order Hidden Markov Model. The data used as input to DER Finder is the base-by-sample coverage matrix. We expect adjacent bases to have high correlation in their coverage values, especially considering the 101-bp read length used in the Y-chromosome experiment presented. To explore the autocorrelation in coverage values across the genome, we estimated the average correlation between base-pairs at increasing distances from each other (Figure 3.1). As expected, the plot displays high correlations between bases that are close together. It also shows that this correlation is close to what would be expected under an au-

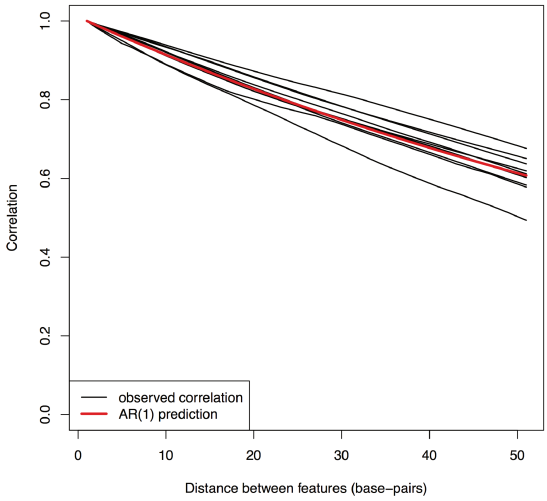

Figure 3.1: Observed average correlation of read coverage (y-axis) between bases of varying distances apart (x-axis), with the predicted AR(1) correlation for this data superimposed in red. Each black line represents one of the nine male samples used in the Y-chromosome analysis in Chapter 2.

toregressive (order 1) model, and such a correlation structure is acceptable under a first-order Markov model. Therefore we believe the first-order model is sufficient for this analysis.

3.2.2 Stationarity and Homogeneity

By default, we assume that D(l) is a non-stationary, homogeneous Markov chain with hidden state probabilities $\pi_d = Pr(D(l) = d)$. However, the user may assume either or both of *stationarity* or *homogeneity*. A stationary Markov chain has constant marginal state probabilities across the genome, and a homogeneous Markov chain has constant transition probabilities between states across the genome. The

HMM also assumes the parameters of the mixture distribution generating the test statistics are the same across the genome. All these assumptions seem reasonable: stationarity can be assumed because we fit the model without using prior information about gene annotations or possible differential expression status, so constant marginal probabilities make sense; we note that we do not assume stationarity in our analysis, since it adds computation time (by adding a constraint to the model) and is not the default in `HiddenMarkov`.[49] Homogeneity is also a reasonable assumption: we ignore prior annotation, so there is no reason to believe transition probabilities should be different in different genomic locations. Finally, along the entire genome, a test statistic high in absolute value should indicate differential expression, while a test statistic low in absolute value indicates no differential expression, so using the same parameters for the test statistics' mixture distribution seems reasonable. If the user is particularly concerned about violations of these assumptions, separate HMMs (with constant marginal probabilities, or varying transition probabilities, or different emission distribution parameters) can be fit on different sections of the genome. Another option is to implement an alternative segmentation algorithm, such as circular binary segmentation.[51]

3.2.3 Test Statistic Distribution

We assume that the test statistic at base l, $s(l)$ has latent state $D(l) = 1$, 2, or 3, and is a draw from a normal distribution, i.e., $s(l) \mid D(l) = d \sim N(\mu_d, \sigma_d^2)$. We

choose this normal distribution because the pre-built functions in the *HiddenMarkov* R package[49] provided the computational framework for fitting this HMM, and because the observed distribution of test statistics seemed well-captured by a normal mixture distribution. As empirical evidence, we consider the test statistics obtained from the Y chromosome analysis presented in Chapter 2 (Figure 3.2): the normal mixture distribution estimated using the process described in Section 3.1.2 seems to fit the observed data quite well. The distribution of the underexpressed statistics overlaps almost entirely with that of the equally expressed statistics, but this is to be expected in Y chromosome data. We also investigated the effect of the prior estimate for π_1, the proportion of base-pairs that are not differentially expressed, using the simulated data described in Section 3.7.1. In that scenario, we set 90% of transcripts to be differentially expressed, and DER Finder produced exactly the same results using $\hat{\pi}_1 = 0.8$, $\hat{\pi_1} = 0.9$, and $\hat{\pi_1} = 0.98$, the latter being the conservative estimate from the two-groups model. In general, we expect DER Finder to be quite robust to choice of parameters for the test statistic distribution: as long as large test statistics are classified as differentially expressed and test statistics close to zero are classified as not differentially expressed in a systematic manner, DER Finder will produce reasonable results.

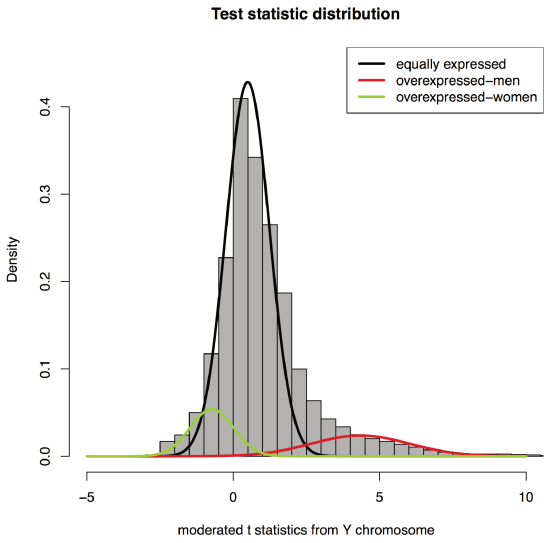

Figure 3.2: Estimated normal mixture distribution of test statistics generated from bases on the Y chromosome. This figure illustrates the plausibility of the assumption that $s(l) \mid D(l) = d \sim N(\mu_d, \sigma_d^2)$. The separate components of the mixture distribution are plotted in different colors.

3.3 Validity of p-value and FDR estimates

DER Finder assignes a measure of statistical significance to each candidate DER using a permutation p-value, as described in Section 2.2.3. Each candidate DER is assigned a test statistic, defined as the mean base-level statistic over all bases contained in the region. To estimate the null distribution of region-level test statistics, permutation is used, and the null distribution is created by pooling null statistics from the entire genome. Using permutation with pooled null statistics is standard practice.[52] It has been demonstrated that strong control of the FDR and FWER are guaranteed when a subset pivotality condition holds.[53] The subset pivotality requires that for any

subset of the null hypotheses, the joint distribution of the p-values for the subset is identical to that under the complete null.[54] This condition holds provided that the p-values under the null hypothesis are jointly uniform.[55] Further justification for our approach is that this type of permutation procedure has been thoroughly studied both empirically and theoretically and is widely applied in the analysis of fMRI data.[56,57]

We show empirically that our null p-values are uniformly distributed and that our estimated FDRs are conservative: using the simulated dataset described in Section 3.7.1, we analyzed all p-values from regions known to contain no differentially expressed bases and found that the distribution was approximately uniform (Figure 3.3). Additionally, the true FDR in the simulation study at a q-value cutoff of 0.05 was 0, meaning our FDR estimate of 0.05 was indeed conservative. A false discovery in this case would be defined as calling a region differentially expressed when it did not overlap a transcript set to be differentially expressed.

We can also use the Y-chromosome experiment to show that the p-values and FDR adjustments used by DER Finder's permutation test are reasonable: for the Y-chromosome data, p-value histograms for each method were created (Figure 3.4). P-values were assigned to each region assigned latent state $D = 2$ by the HMM step in DER Finder, to each transcript in Cufflinks, and to each exon in EdgeR and DESeq. The observed distributions were shaped as expected in the results from DER Finder, EdgeR, and DESeq: in the comparison between sexes, many low p-values were observed, corresponding to the fact that most of the Y chromosome should be

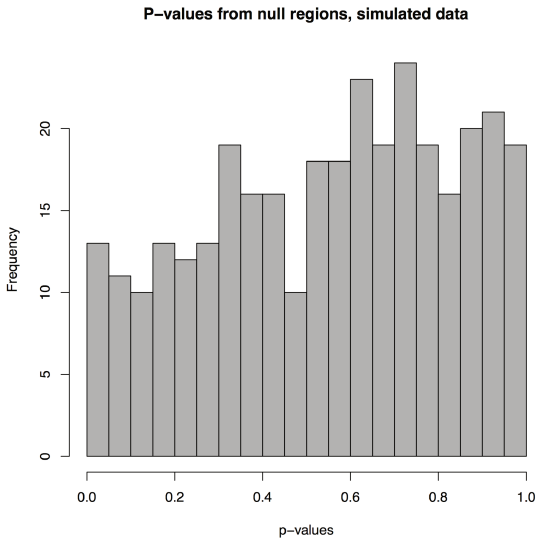

Figure 3.3: Histogram of null p-values from a small simulation study, where a region is considered null if none of the bases in that region were contained in a transcript that was set to be differentially expressed. This distribution is approximately uniform, which implies that these p-values have good theoretical properties.

differentially expressed. However, based on the p-value histogram generated from the Cufflinks transcripts, the analysis of differential expression between sexes did not produce many small p-values. Instead, it produced a cluster of p-values between 0.2 and 0.4, which is an unexpected finding given the nature of Y chromosome expression differences between males and females. This simple analysis shows that the statistical methodologies used by DER Finder, EdgeR, and DESeq produce reasonable results on an easy problem, while Cuffdiff exhibits problems even in a very simple scenario.

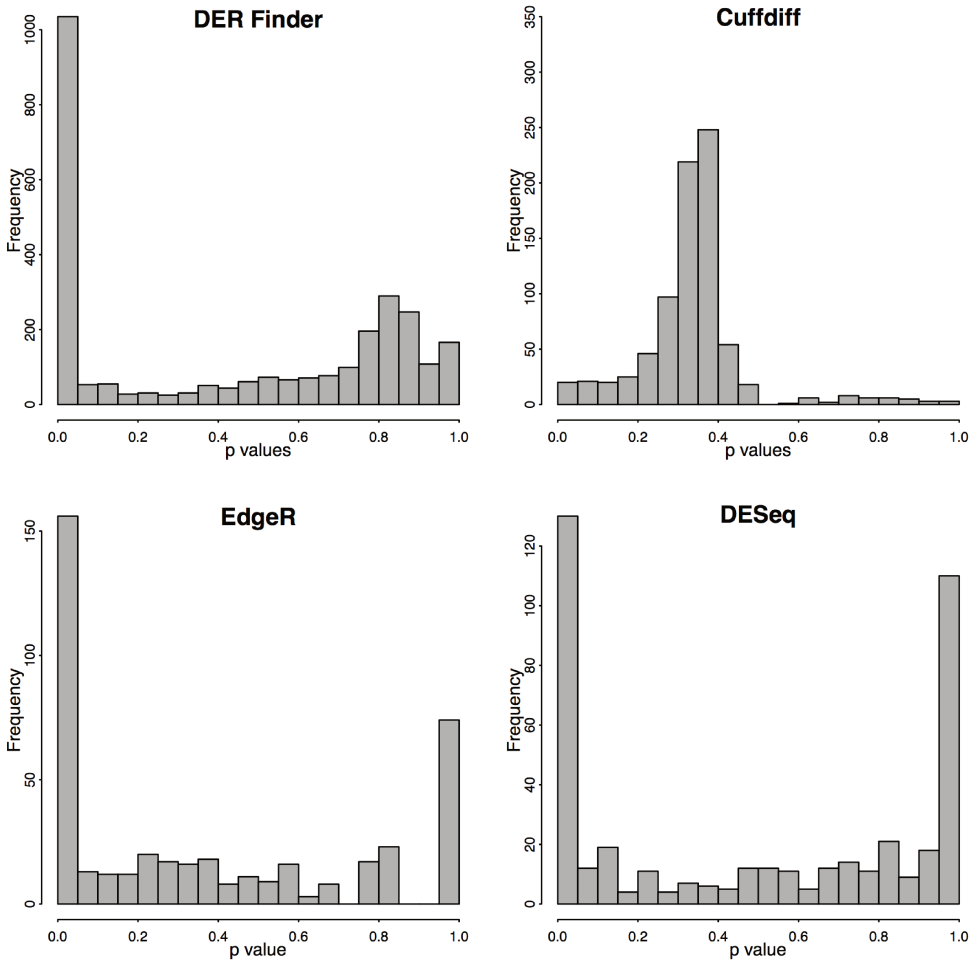

Figure 3.4: P-value histograms for tests of differential expression on the Y chromosome between males and females. For all methods except Cuffdiff, substantial differential expression is evident in the comparisons between sexes, as expected. The Cuffdiff p-value distribution is quite unusual and indicates that using p-values adjusted for multiple testing to assess significance may be problematic.

3.4 Details for Y chromosome experiment

Section 2.3 presents an experiment in which we compared male and female gene expression on the Y chromosome. The data consisted of unpaired, 101-bp RNA-seq reads from 15 control samples (9 male, 6 female) of postmortem brain tissue. These reads were aligned to the Ensembl GRCh37 genome[58] using TopHat version 2.0.8 with default parameters, which allow mutiple alignments per read to be reported. DER Finder's coverage matrix was calculated based on these TopHat alignments. Results from DER Finder were compared to results from the Cufflinks/Cuffdiff pipeline, EdgeR, and DESeq. EdgeR and DESeq analyses were run at the exon level, using exon-by-sample count tables created based on the TopHat alignment file with RSamtools[59] and GenomicRanges.[60] Exon-level expression summaries for EdgeR and DESeq were calculated using the `summarizeOverlaps` function in GenomicRanges, using the union model to count reads falling within overlapping exons (`mode="Union"` option in `summarizeOverlaps`). Default parameters and library size adjustments were used in EdgeR and DESeq. For these exon-by-sample count tables and for determining DER Finder's regions' overlaps with exons, we considered all annotated exons in the Ensembl GRCh37 build, as annotated in the databases used by the biomaRt Bioconductor package.[61]

For Cufflinks/Cuffdiff, we used Cufflinks version 2.0.2 for transcript assembly and Cuffdiff version 2.0.2 for differential expression analysis. Default parameters were

used for both steps.

The main model for DER Finder (model 2.1) was fit as follows: g was defined as the function $g(x) = log_2(x + 32)$, $X_{2i} = 1$ if sample i was male and 0 if sample i was female (this is essentially the case/control scenario, so $P = 2$), and W_{i1} was defined as the median of nonzero coverage values for each sample. Using this model setup, $\hat{\beta}_2(l_j)$ represents the estimated log (base 2) fold change in expression of base l_j for males compared to females, when all coverage matrix values are offset by 32 to ensure that zero counts do not cause problems in the log transformation. No other confounders were included in the analysis. The test statstic on the base level was *limma*'s moderated t statistic (see Section 3.1.1), and the HMM with DER Finder's default parameters (see Section 3.1.2) was run on these t statistics to obtain candidate DERs. To obtain p-values for the candidate DERs, a permutation test was run as described in Section 2.2.3, using $B = 10$ permutations. All p-values (from all pipelines) were adjusted for multiple testing by controlling the false discovery rate, so the q-value[52] was used as a measure of statistical significance.

To connect the results from this experiment to annotated features, we labeled each DER with what type of genomic event it might indicate and which annotated features are involved (Table 3.1). These labels aid in determining which exons and genes are showing differential expression signal and finding regions that may indicate phenomena like possible alternative splicing. Further exploration of these regions is possible using assemble-then-annotate methods to evaluate potential alternative

Result	**Flag**
A set of regions of state $D = 2$ overlaps more than 80% of an annotated exon.	Differentially Expressed Exon
There exists a set of regions of state $D = 2$ with differentially expressed exon flags such that all exons in a given gene are flagged by the set	Differentially Expressed Gene
There exists a set of regions of state $D = 2$ with differentially expressed exon flags such that at least one, but not all, of the exons in a given gene are flagged by the set	Unknown Event of Interest (e.g., alternative splicing)
Region of state $D = 1$ does not overlap any annotated exons	Novel Transcribed Region
Region of state $D = 2$ does not overlap any annotated exons	Novel Differentially Transcribed Region

Table 3.1: Possible genomic events indicated by results from DER Finder

or differential splicing events. Due to variance in read coverage across the genome, we observed some regions shorter than the length of an individual read. These small regions are particularly detrimental in the annotation and labeling step. We therefore choose to disregard regions shorter than the read length. Regions flanking very short ($<$ 5-base) transitions between states are merged.

3.5 Additional figures illustrating problems with annotate-then-identify methods

Figure 2.3 illustrates specific instances in the analysis of the human Y chromosome where DER Finder correctly identifies differential expression between sexes and EdgeR and DESeq do not, either because an exon was incorrectly annotated

or because the differential expression did not occur within an exon at all. The instances shown in the text are not isolated: in fact, 280 non-exonic regions of the Y chromosome were identified by DER Finder as significantly differentially expressed ($q < 0.05$). Additionally, Figure 3.5 demonstrates that differential expression does not always occur within exon boundaries, so an identify-then-annotate method may be necessary to achieve high sensitivity. We examined differentially expressed Y-chromosome regions found by DER Finder that overlapped only part of an exon: for a fixed percentage x, we gathered the DERs (identified with DER Finder) that overlapped no more than $x\%$ of an exon. Then we calculated the fraction of the set exons overlapped by those DERs that EdgeR and DESeq called differentially expressed ($q < 0.05$). Figure 3.5 plots different values of x against these fractions. The figure's message is that many exons showing a differential expression signal when analyzed with DER Finder are not called differentially expressed by EdgeR and DESeq, even in an easy analysis of differential expression between males and females on the Y chromosome. Figure 2.3 is a specific example of this problem, and Figure 3.5 suggests that the issue is not confined to only one example.

Figure 3.6 shows how DER Finder's agreement with EdgeR and DESeq's findings changes based on how much of an exon we require a DER to overlap in order to call that exon differentially expressed. While Figure 3.5 looked at how much EdgeR and DESeq agreed with DER Finder's results, this figure examines how much DER Finder agrees with EdgeR and DESeq's results. Given a percentage x to use as a cutoff for

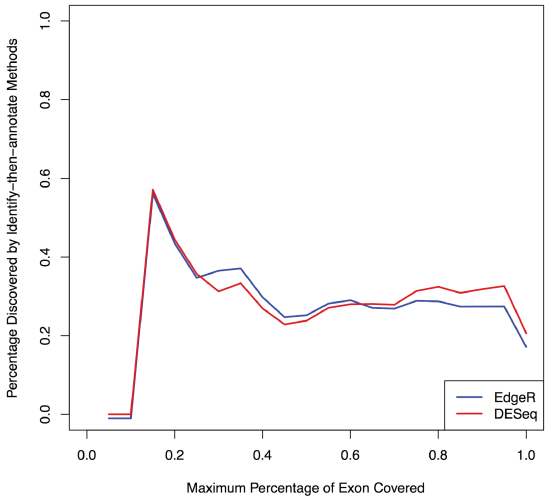

Figure 3.5: Percentage of the exons overlapped by no more than $x\%$ (for varying values of x) of a differentially expressed region ($q < 0.05$) from DER Finder that are also identified as differentially expressed ($q < 0.05$) by EdgeR and DESeq. (The EdgeR line was lowered by 0.01 so the differences between the two lines on the left side of the plot would be visible.)

how much an exon must be overlapped by a DER in order to be called differentially expressed by DER Finder, Figure 3.6 shows how many of the exons identified by DESeq or EdgeR as differentially expressed are also $x\%$ covered by a DER. As more overlap is required for a differential expression call, the percent agreement between DER Finder and the identify-then-annotate methods decreases, but overall, most of the exons identified by EdgeR and DESeq also show a signal in DER Finder.

Together, Figures 3.5 and 3.6 show that DER Finder identifies most of the differential expression found by EdgeR and DESeq, but the identify-then-annotate methods miss signals identified by DER Finder due to the heavy reliance on pre-specified exon annotation.

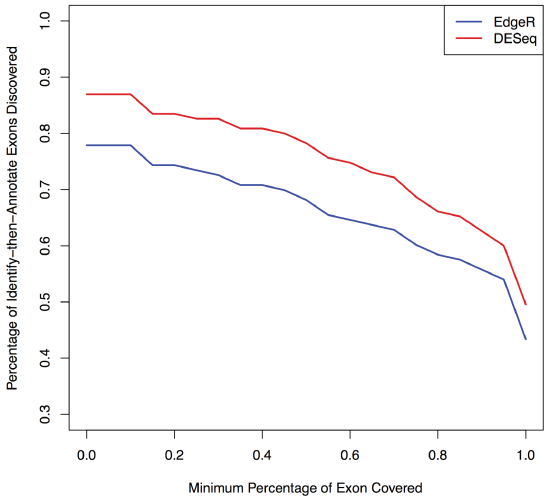

Figure 3.6: Percentage of exons called differentially expressed ($q < 0.05$) by EdgeR and DESeq that are overlapped by at least $x\%$ of a differentially expressed region ($q < 0.05$) from DER Finder, for varying values of x.

3.6 Additional Y-chromosome analysis: agreement between methods

To determine the extent to which the different pipelines called the same features differentially expressed, we quantified differential expression and overlap between findings at varying q-value cutoffs, comparing DER Finder to Cufflinks/Cuffdiff (Table 3.2) and DER Finder to EdgeR and DESeq (Table 3.3). DER Finder produces better results than Cufflinks/Cuffdiff: we find differential expression between males and females on the Y chromosome, and find no differential expression between the males, while Cufflinks does not find differential expression between sexes unless the q-value

cutoff is above 0.45. When the q-value cutoff is high (0.50), only 5.3% of the differentially expressed Cufflinks transcripts are also called differentially expressed by the new method: as expected, transcripts with high q-values are not overlapped by differentially expressed regions from the new method (regions that are equally expressed will not make it past the HMM segmentation step). On the other hand, 32.5% of the differentially expressed regions ($q < 0.50$) are overlapped by differentially expressed transcripts, which shows some agreement between the methods.

The comparison to EdgeR and DESeq shows the annotation-based results to be somewhat similar. The q-value cutoff did not seem to matter when assessing exon-specific results from DER Finder for the male-to-female comparison of the Y chromosome (Table 3.3). Overall, the DER Finder results and the EdgeR and DESeq results were somewhat comparable on the exon level. The q-value cutoff had no bearing on the DER Finder results: all the differentially expressed regions covering at least 80% of an annotated exon had small q-values. At low q-values, DER Finder identifies more exons as differentially expressed than EdgeR and DESeq do, with some agreement between all three methods. Results from Table 3.3 are from the comparison between sexes; the males showed no differential exon expression in EdgeR/DESeq (all but two q-values 1), or DER Finder (minimum q-value of 0.86). It is worth noting that the method of summarizing the number of reads per exon affects EdgeR and DESeq results: in particular, the common counting methods do not allow reads to be counted toward more than one feature, so overlapping exons do not usually get

any reads assigned to them at all. In our experiment, this led to 345 exons having overlapping DERs assigned to them but not even being tested by EdgeR or DESeq. This issue explains some of the discrepancy between the exon-level findings for DER Finder and EdgeR/DESeq. Also, though DER Finder identifies more exons overall as being differentially expressed, 54 exons are identified only by EdgeR or DESeq. Closer examination of the coverage patterns of these exons revealed that most of them were either (a) very lowly expressed overall, or (b) were less than 80% covered by DERs, so the exons themselves were not called differentially expressed because of the cutoffs defined in Table 3.1. Users can adjust DER Finder parameters if they are particularly interested in discovering differential expression of lowly-expressed features (e.g., the function $g()$ chosen in model 3.1 could be $g(x) = log_2(x+0.5)$ rather than $log_2(x+32)$, which is what was was used the Y chromosome comparison). Also, DER Finder generally does show at least some signal in the general area of the exons in question, even if that signal does not overlap the exon by 80%, so the results still give meaningful information. Overall, these findings confirm the result that EdgeR, DESeq, and DER Finder perform similarly when analyzing already-annotated features.

3.7 Experimental Design Concerns

Biologists who collect RNA-seq data must make several decisions when designing their experiments. Two important considerations are whether to use single-end or paired-

q-value	**# DE regions**	**# DE transcripts**	**# agreeing regions**	**# agreeing transcripts**
		(a) males vs. females		
0.05	534	0	NA	0
0.10	1009	0	NA	0
0.50	1185	758	40	385
0.80	1259	787	48	412
		(b) males vs. males		
0.05	0	0	NA	NA
0.10	0	0	NA	NA
0.50	0	0	NA	NA
0.80	0	458	0	NA

Table 3.2: Comparison of results from DER Finder to Tophat-Cufflinks-Cuffdiff. The first column is the number of differentially expressed regions found by DER Finder, while the second column is the number of differentially expressed transcripts found by Cufflinks, both at the specified q-value cutoff. The third column shows how many of the differentially expressed Cufflinks transcripts are at least 80% overlapped by a differentially expressed region from DER Finder, while the fourth column shows how many of the differentially expressed regions are at least 80% overlapped by a differentially expressed Cufflinks transcript.

q-value	# DE Regions	# DE DER Finder exons	# DE EdgeR exons	# DE DESeq exons	DER Finder /EdgeR overlap	DER Finder /DESeq overlap	EdgeR /DESeq overlap	All overlap
0.05	534	411	113	115	66	76	97	65
0.10	1009	417	125	120	76	81	106	74
0.50	1185	417	143	165	80	86	127	79
0.80	1259	417	153	187	83	89	134	82

Table 3.3: Comparison of results from DER Finder to EdgeR and DESeq, analyzing differential expression at the exon level on the Y chromosome between males and females. The first column is the number of differentially expressed regions found by DER Finder, and the second, third, and fourth columns are the number of differentially expressed exons found by each method at the specified q-value cutoff. Differentially expressed exons for DER Finder were defined as exons that were more than 80% covered by regions of state $D = 2$; the q-value for each exon was taken to be the q-value of the region most overlapping it. The last four columns show the number of exons found by two or all three methods.

end reads and how deeply to sequence the samples. We address these considerations and their impact on DER Finder's results in this section, using a small simulation study to support the conclusions drawn.

3.7.1 Simulation set-up

A small, 20-sample RNA-seq dataset with pre-defined differential expression was simulated using Flux Simulator version 1.2.[62] We simulated 76-bp paired-end reads from 1000 randomly selected transcripts on chromosome 22. For these 1000 transcripts, we simulated approximately 400,000 reads per sample. We then randomly chose 50 of these transcripts to be overexpressed in 10 of the samples (group A) and 50 different transcripts to be overexpressed in the other 10 samples (group B). Overexpression was

simulated by generating an additional 80,000 reads from the designated 50 transcripts for each sample. Essentially, this process mimicked a 5x fold change per overexpressed transcript. The default error model for 76-bp reads was utllized, and all other parameters were left at the default value. The command run for each simulated sample was `flux-simulator -t simulator -x -l -s -p sample.par`. An example parameter (.par) file is available on GitHub (`https://github.com/alyssafrazee/derfinder`). The simulated reads from each dataset were aligned to the Ensembl GRCh37 genome[58] using TopHat 2.0.8 with default parameters, and coverage matrices were created from the TopHat alignment file.

3.7.2 Paired-end data in RNA-seq analysis

It is well-known that using paired-end data, i.e., data consisting of reads from both ends of the mRNA fragments instead of just one end, is better than using single-end data, even though paired RNA-seq experiments can cost up to twice as much as a single-end experiment.[14,63] Mate-pair information is used during read alignment to more accurately determine the reads' best mappings. Paired-end data is especially important in assemble-then-identify methods because it yields more reliable transcript assemblies and better per-transcript abundance estimates. Because annotate-then-identify and identify-then-annotate methods do not involve assembly or transcript-level quantification, paired-end data only improves these methods inasmuch as it improves the read alignment step. Therefore, since read alignment can be done with

either single-end or paired-end reads, it is appropriate to use DER Finder with either type of data. The coverage matrix would be calculated the same way for paired data as it is for single-end data; each mate of a mate pair would contribute a coverage value of 1 to all the bases to which it aligns.

The Y-chromosome analysis in this manuscript was done using single-end data, which may put Cufflinks/Cuffdiff (the assemble-then-identify method) at a disadvantage when comparing it to the other tools. To determine whether the poor statistical results from Cufflinks/Cuffdiff were due to the single-end reads, we ran Cufflinks and Cuffdiff (version 2.0.2, with default parameters) on the simulated dataset. Even though this dataset was paired-end and contained transcripts known to be highly differentially expressed, the statistical results from Cufflinks/Cuffdiff were unreasonable: they did not reflect any differential expression (Figure 3.7b), while DER Finder did detect the signal (Figure 3.7a). Therefore, we contend that while paired-end data may improve assembly methods, it is not the deciding factor in whether the Cufflinks/Cuffdiff pipeline produces reasonable statistical results.

3.7.3 Effect of sequencing depth

Sequencing depth (or read coverage) refers to how many times each mRNA nucleotide in the sample is read by the sequencing machine. Experiments with greater sequencing depth are better able to detect expression differences for features that are lowly expressed overall. This property holds for most existing differential expression anal-

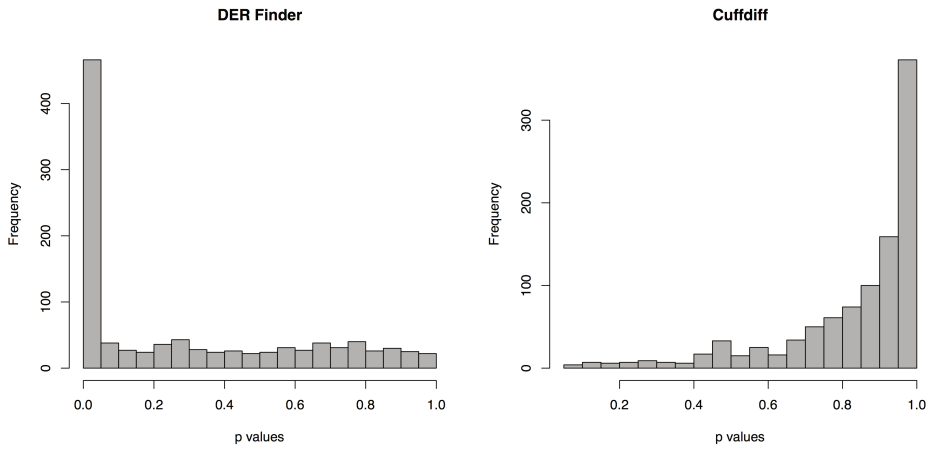

Figure 3.7: P-value histograms from a small, paired-end simulation study with known differentially expressed transcripts. DER Finder's p-values have the expected distribution, while Cuffdiff produces unreasonable statistical results, calling nothing differentially expressed (minimum q-value 0.999) despite 10% of transcripts being overexpressed (fold change = 5) in one condition. This figure demonstrates that paired-end sequencing does not eliminate the problems with Cuffdiff's statistical analysis.

ysis methods, including DER Finder. Therefore, experimenters wishing to use DER Finder and detect differential expression for lowly expressed features should deeply sequence their samples. One specific consideration in DER Finder is the choice of $g()$ in model 2.1. In general, we recommend using $g(x) = log_2(x + k)$, where k is a constant that allows the method to avoid taking the log of 0. In our experiment, we set $k = 32$ because we were not particularly interested in differential expression in areas with low coverage, and offsetting all counts by 32 attenuates the fold changes observed in very low-coverage regions. However, if the sequencing depth is high, the user may want to increase k (if lowly-expressed features are not of interest), since the method will be more sensitive to differential expression of lowly-expressed features

with deep sequencing. Similarly, if the samples are not sequenced very deeply, the user may want to decrease k, since true differential expression may not be detected if the samples' coverage values are offset too much. A good choice for a small k is $k = 1$, since zeros in the original data will still be zeros after log-transforming.

We also investigated the effect of sequencing depth using the simulated dataset described above in addition to two more simulated datasets. These additional datasets were generated in the same manner as the first dataset except for read coverage: the first additional dataset had half as many reads as the original dataset, and the second had 1/4 as many reads as the original dataset. Based on the median length of the transcripts included in these experiments, the coverages for these datasets were approximately 24x, 12x, and 6x, respectively. Coverage matrices were created using TopHat alignments, and DER Finder was run on the chromosome 22 coverage matrix for each dataset, with model 2.1 defined as follows: $g(x) = log_2(x + 32)$, $X_i = 1$ for samples in group A and 0 for samples in group B, and W_{i1} was set as the median nonzero coverage value for each sample.

In this simulated dataset, DER Finder using the 24x and 12x datasets found 435 and 433 differentially expressed regions, respectively ($q < 0.05$), while the 6x dataset did not find any differential expression (minimum q-value 0.18). This is consistent with what we expect: the same offset ($k = 32$) was used for all three datasets, and this appears to be too much of an offset for the low-coverage (6x) dataset. To further investigate these findings, we used varying q-value cutoffs to create ROC curves for

the different coverage levels (Figure 3.8). DER Finder appears to be performing well in terms of sensitivity and specificy for the 12x and 24x experiments: for example, in the 24x experiment, 97 out of the 100 pre-set differentially expressed transcripts were overlapped by a significant ($q < 0.05$) DER, while 93% of the transcript features (exons, etc.) that were not simulated as differentially expressed were overlapped by regions in the equally expressed state or with $q \geq 0.05$. In general, there was very little difference between 12x and 24x coverage in this simulation, but 6x read coverage appears to be too shallow when the offset is set at 32.

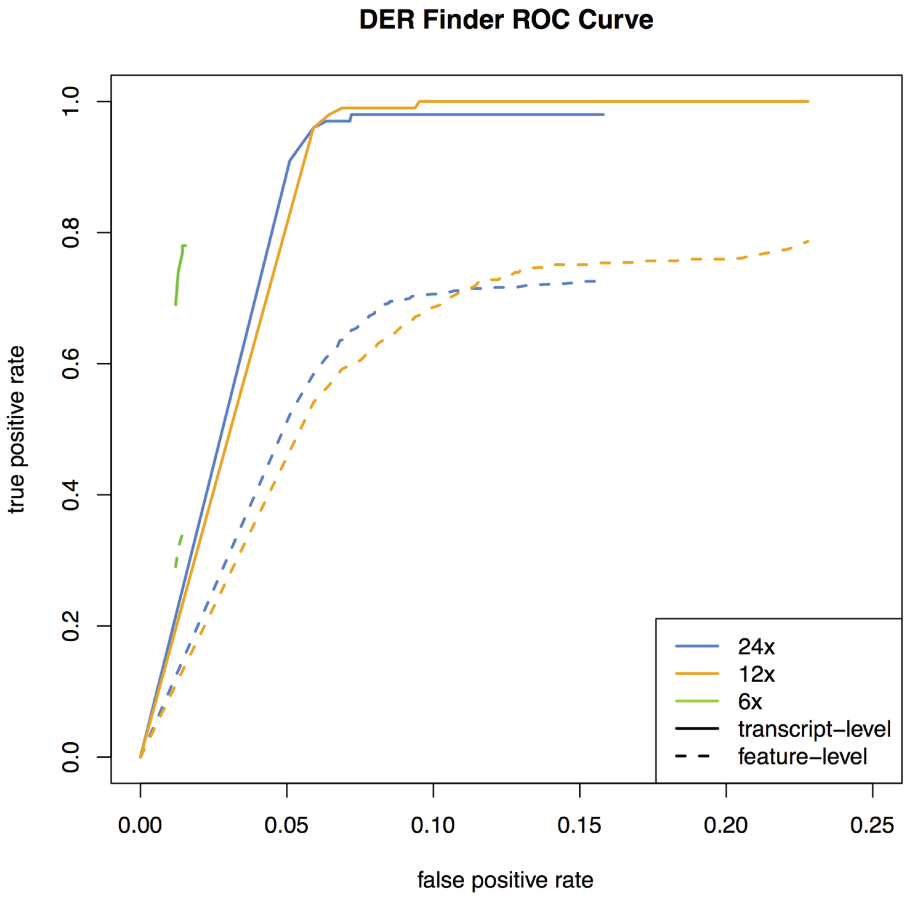

Figure 3.8: ROC curves from DER Finder, created based on the simulation study with known differential expression. Sequencing depth is noted by color, while line type denotes different ways of determining differential expression calls: the dashed lines were created at the feature level, i.e., the true positive rate was the percentage of differentially expressed transcript *features* (exons, etc.) that were overlapped by a significant DER. The solid lines were created at the transcript level, i.e., the true positive rate was the percentage of *transcripts* with at least one feature overlapped by a significant DER. DER Finder is performing well in terms of sensitivity and specificity when the sequencing depth is sufficient.

Chapter 4

Bridging the gap between transcriptome assembly and expression analysis

4.1 Introduction

This chapter describes work published in separate form in the journal *Nature Biotechnology*, with contributions from co-authors Geo Pertea, Andrew E. Jaffe, Ben Langmead, Steven L. Salzberg, and Jeffrey T. Leek.

Analysis of raw RNA sequencing reads makes it possible to reconstruct complete gene structures, including multiple splice variants, without relying on previously-established annotations.[13,22,64] Downstream statistical modeling of summarized gene

or transcript expression data output from these pipelines is facilitated by the Bioconductor project,[25] which provides open-source tools for analysis of high throughput genomics data. However, the outputs of upstream processing tools are often aggregated across samples, making it difficult to estimate between-sample variability, or are not in a format that is readily compatible with downstream Bioconductor packages. This gap has prevented rigorous statistical analysis of transcript-level data: some two-group analysis was possible, but complex analyses like eQTL (expression quantitative trait loci), timecourse studies, investigation of the effect of continuous covariates on expression, and adjustment for confounders were not. These difficulties have led to considerable controversy in the analysis of population-level RNA-seq data.[65] Here and in Chapter 5, we report the development of two pieces of software, Tablemaker and Ballgown, that bridge the gap between transcriptome assembly and fast, flexible differential expression analysis (Figure 4.1).

First, Tablemaker uses a GTF file (the standard output from any transcriptome assembler) and spliced read alignments to produce files that explicitly specify the structure of assembled transcripts, mappings from exons and splice junctions to transcripts, and several measures of feature expression, including FPKM (Fragments Per Kilobase of transcript per Million reads sequenced)[13,22] and average per-base coverage (Section 5.1). Tablemaker wraps Cufflinks to estimate FPKM for each assembled transcript. After the transcriptome assembly is processed using Tablemaker, the output files (Section 5.1) can be explored interactively in R using the Ballgown

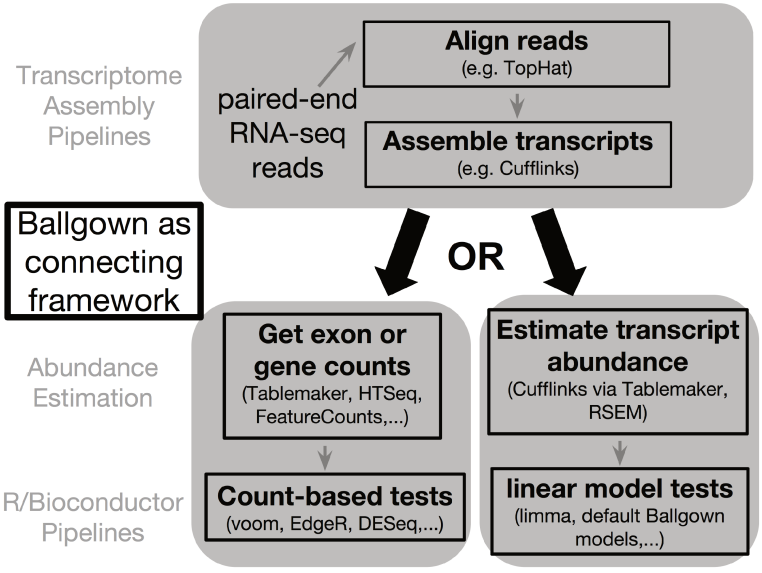

Figure 4.1: The Ballgown pipeline. Ballgown is designed to be a tool-agnostic bridge between transcriptome assemblers and abundance estimation tools, and fast, flexible differential expression analysis pipelines in R and Bioconductor. Ballgown as a bridge between transcriptome assembly and fast, flexible differential expression analysis. For example, the Ballgown workflow connects transcript assembly tools like TopHat and Cufflinks to Bioconductor tools like EdgeR and DESeq for downstream analysis, but it is not specific to these particular tools. The software can be used with any assembly whose structure is specified in GTF format, coupled with a set of spliced read alignments in BAM format. RSEM and StringTie (in addition to Cufflinks) are currently officially supported, and we plan to add support for more tools.

package. Ballgown converts Tablemaker's assembly structure and expression estimates into an easy-to-access R object (Figure 4.2) for downstream analyses. Tablemaker is specifically designed to connect the Cufflinks assembler with downstream R tools, but the Ballgown R object can be created with transcriptomes from any assembler with output in the right format. Currently, StringTie,[66] a new, efficient assembler, can output the proper files, and Ballgown is also compatible with RSEM's `rsem-calculate-expression`[67] abundance estimation tool. Ballgown can be used to visualize the transcript assembly on a gene-by-gene basis, extract abundance estimates for exons, introns, transcripts, or genes, and perform linear-model-based differential expression analyses (Section 5.2.3).

The basic linear modeling strategy for differential expression testing implemented in Ballgown allows for analysis of eQTL, timecourse, continuous-covariate, or confounded experimental designs at the exon, gene or transcript level. This approach is similar to the linear modeling strategy implemented in limma,[46] without empirical Bayes shrinkage, and can be applied to exon or gene counts available through the Ballgown object after appropriately transforming the count data.[68] Alternatively, users can apply widely used Bioconductor packages for sequence count data.[27,29] There is no existing statistical software that allows this level of flexibility for modeling transcript-level expression data. Count-based modeling strategies are not applicable to transcript-level data[69] and Cuffdiff2 can only be applied to two-group transcript-level differential expression analysis.[63] EBSeq could be used in combina-

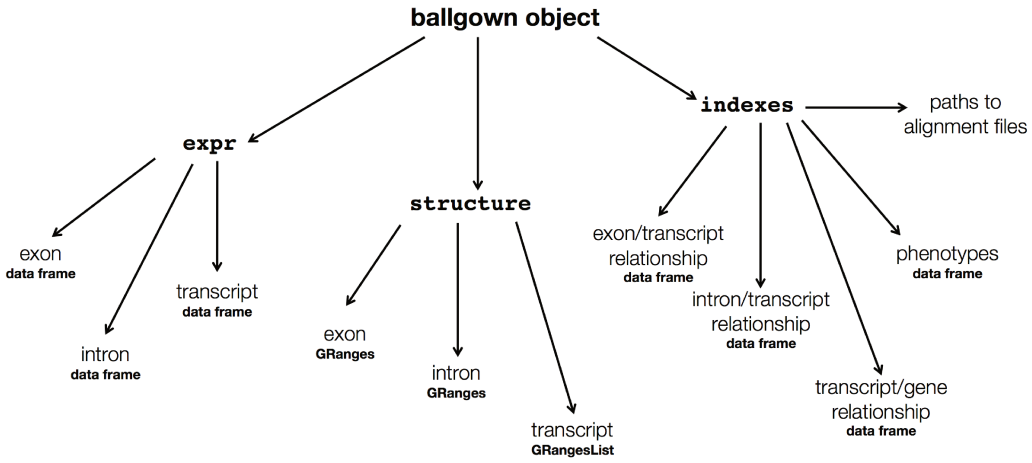

Figure 4.2: The `ballgown` data structure. The Ballgown R provides a comprehensive data structure for transcriptome assemblies. The package loads assembly data into an object with linked data frames of expression measurements (`expr`) for exons, introns, and transcripts. The object also loads information about exon, intron, and transcript structures (`structure`), utilizing the efficient GenomicRanges[60] data structures for storage. Finally, the object contains other relevant assembly data (`indexes`), including phenotype data, relationships between exons, introns, and transcripts, and paths to alignment files on disk for easy connection with the assembly.

tion with RSEM as a pipeline for transcript-level differential expression analysis, but it less efficient than linear modeling and does not handle experimental designs beyond multi-group comparison.[70]

In this chapter, we illustrate how to use Tablemaker and Ballgown with Tuxedo suite, a widely used pipeline for transcript assembly, quantification, and flexible differential expression analysis at transcript resolution. The Tuxedo suite consists of aligning reads using Bowtie[71] and TopHat,[39] assembling transcripts using Cufflinks,[22] and differential expression analysis using Cuffdiff2.[24] This suite has been used in many projects,[72–74] including the ENCODE[75] and modENCODE[76] consortium projects. However, statistical analysis through Cuffdiff2 can only be applied to two-group differential expression analyses, is computationally demanding, and produces strongly conservative estimates of statistical significance. While several other fast and accurate tools for differential expression analysis such as EdgeR,[27] DESeq,[29] and Voom[68] are present in Bioconductor,[25] there is no software that connects these tools to the estimated transcript structures and abundances output by tools such as the Tuxedo suite. Further, per-feature read counts are not appropriate for isoform-level analysis, since isoforms from the same gene may have a high degree of overlap leading to ambiguous read counts. Here we integrate the Tuxedo suite with Tablemaker, Ballgown, and downstream Bioconductor packages to improve statistical accuracy, flexibility in experimental design, and computational speed if isoform-level of RNA-seq analyses.

4.2 Statistical Accuracy

4.2.1 Negative Control Experiment

First, we show that the default methods in Ballgown can work in the absence of a differential expression signal. For this analysis, we downloaded and processed data from the GEUVADIS RNA sequencing project[77,78] (Section 5.3). After aligning RNA-seq reads, assembling the transcriptome, and processing the results with Tablemaker, we used Ballgown to load the data into R, where we extracted a single-population subset of data to study. The populations included in the GEUVADIS study were Utah residents with Northern and Western European ancestry (CEU), Yoruba in Ibadan, Nigeria (YRI), Toscani in Italy (TSI), British in England and Scotland (GBR), and Finnish in Finland (FIN). Considering only individuals in the FIN population (n = 95), we randomly assigned subjects to one of two groups and tested all assembled transcripts for differential expression between those two groups. We compared results from using linear models (Ballgown), Cuffdiff2,[24] and EdgeR[27] (at the exon level). We used transcript FPKM as the transcript expression measurement in Ballgown, and we considered per-exon read counts for EdgeR. In this type of experiment, the distribution of the p-values from all the transcripts should be approximately uniformly distributed, and q-values[52] should be large.

As expected, the transcript-level p-values from the linear model tests implemented

in Ballgown were arguably approximately uniformly distributed (Figure 4.3), though there are fewer p-values near 0 than we might expect, indicating that the test is potentially slightly conservative. All transcripts had q-values of approximately 1, indicating that these models do not generate excess false discoveries. We compared this result to the statistical results from Cuffdiff2 (version 2.2.1, the newest release available as of August 2014) on the same dataset and found that the p-values obtained using Cuffdiff2 were not uniformly distributed, but that the distribution had more mass near 1 than near 0 (Figure 4.4a). This indicates that Cuffdiff2 may be conservatively biased and calls into question the use of the q-value as a multiple testing adjustment, since it assumes uniformly-distributed p-values. Finally, at the exon level, EdgeR called two exons differentially expressed ($q < 0.05$) , and the exon-level p-value distribution was not uniform, having a bit of extra mass around 0.1 (Figure 4.4b). These results show that using a well-established, count-based method gives a too-liberal result, that Cuffdiff2 is likely conservatively biased, and that using a linear model test like the one implemented in Ballgown gives a reasonable p-value distribution without calling any transcripts differentially expressed.

The linear models from Ballgown took 18 seconds to run on a standard laptop (MacBook Pro, 8G memory). For comparison, Cuffdiff2 took 69 hours and 148G of memory using 4 cores on a cluster node. EdgeR was also run on the laptop and took 2.5 minutes.

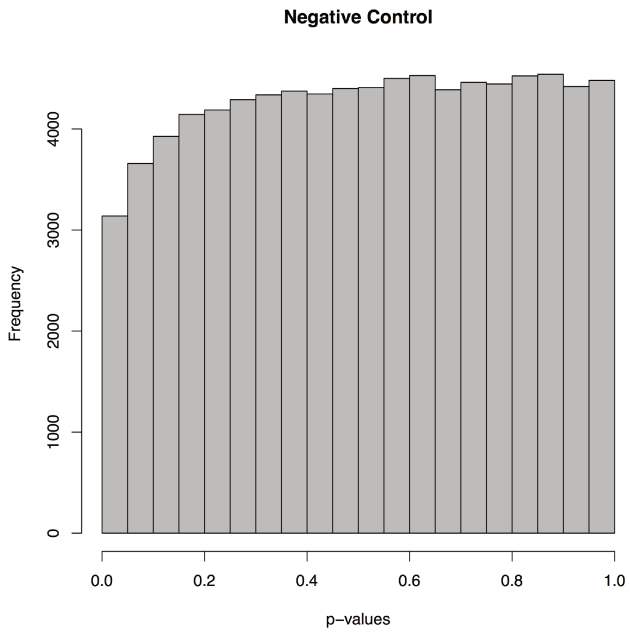

Figure 4.3: Distribution of transcript-level p-values obtained with Ballgown's F-tests in an experiment without signal. This distribution is close to uniform, with slightly fewer small p-values that we might expect under true uniform, but indicates the tests are performing reasonably compared to the methods whose p-value distributions are illustrated in Figure 4.4.

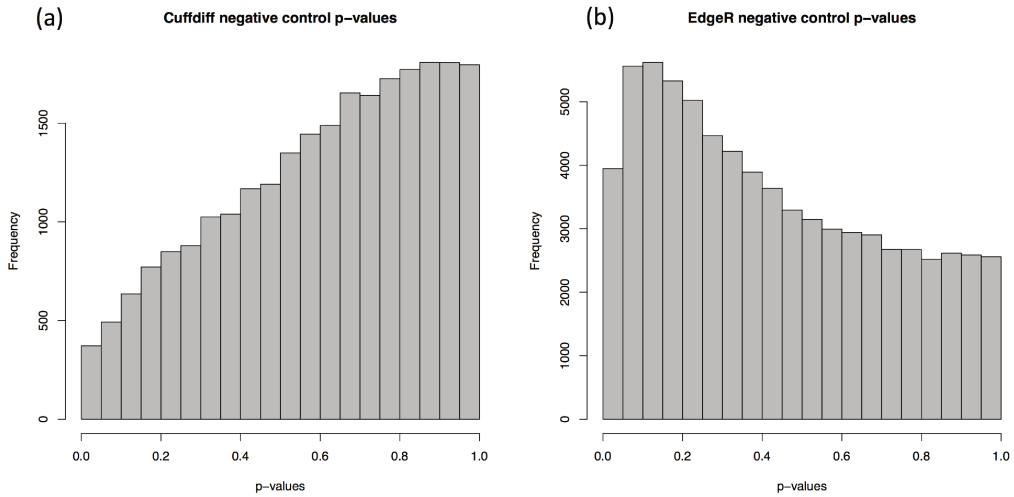

Figure 4.4: P-value histograms of results of differential expression analyses between two randomly selected groups: Cuffdiff and EdgeR. Differential expression results from Cuffdiff (version 2.2.1, the newest release available as of August 2014) in an experiment without signal gave p-values that were not uniformly distributed but instead were biased toward 1 (Panel a). At the exon level, the p-value distribution from EdgeR was also not uniform, having a bit of extra mass around 0.1 (Panel b). These results show that a well-established, count-based methods gives a slightly too-liberal result on this kind of experiment and illustrates a potential conservative bias still present in Cuffdiff version 2.2.1.

4.2.2 Positive Control Experiment

The negative control experiment showed Ballgown's default statistical tests are appropriately conservative when there is no signal present in the data. We carried out a second experiment to investigate whether default statistical tests are capable of making discoveries when differential expression is present. For this experiment, we analyzed differential expression of Y-chromosome transcripts between males and females, a test dataset in which all transcripts should be differentially expressed. We used a dataset consisting of the 95 FIN individuals (58 females, 37 males) in the GEUVADIS RNA-seq dataset (Section 5.3). The p-value histogram from this experiment using the linear model framework implemented in Ballgown shows a very strong signal (Figure 4.5a). Of the 433 assembled transcripts on the Y chromosome, 225 had a mean FPKM greater than 0.01 in the males. 58% of these 225 transcripts were called differentially expressed with a q-value less than 0.05 and 72% with a q-value less than 0.2. This result shows that the models in Ballgown are capable of discovering true signal in the dataset.

The p-value histogram for the latest Cuffdiff2 version (2.2.1) also revealed a signal (Figure 4.5b), which is an improvement compared with earlier versions of Cuffdiff2 on similar Y-chromosome tests[79] (Figure 3.4b). However, only 29 of the 433 assembled transcripts were tested using the inclusion criteria implemented in Cuffdiff2. Of the 29 tested, 24 had $q < 0.05$ and 26 had $q < 0.2$. This suggests that Cuffdiff2 is too

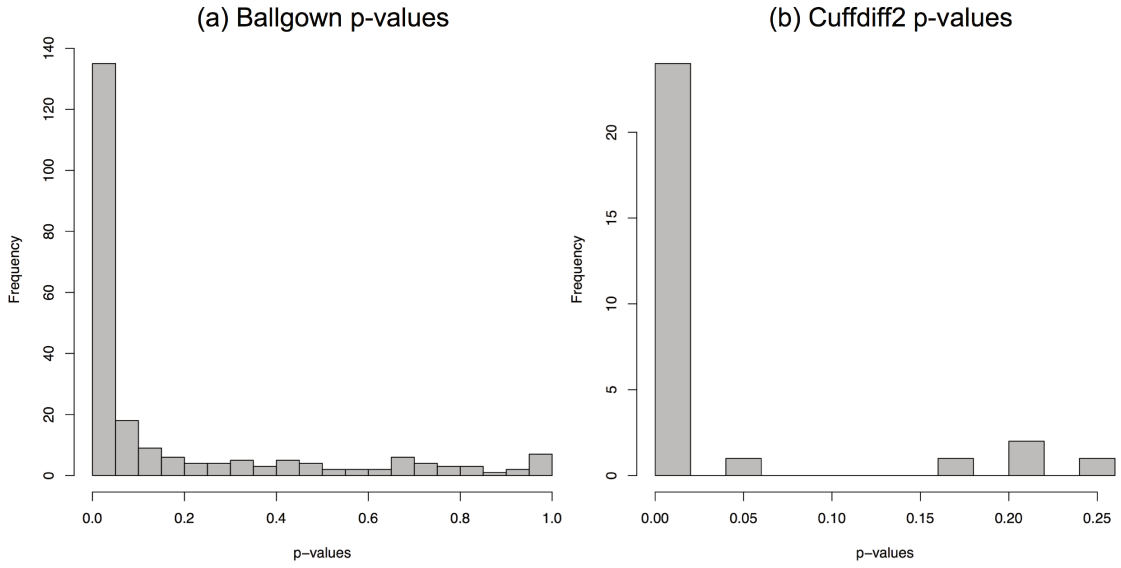

Figure 4.5: P-value histograms of results of differential expression analyses between males and females, for Y-chromosome transcripts. Panel (a) shows the transcript-level p-value distribution from Ballgown F-tests; Panel (b) shows transcript level p-values from Cuffdiff 2.2.1. Both show a strong signal, as expected, but Cuffdiff2 is conservative in terms of the number of transcripts it tests.

conservative to detect appropriate levels of differential expression in this experiment. The Y chromosome linear models from Ballgown took less than 0.1 seconds to run after Tablemaker, whereas Cuffdiff2 took 58 hours and 178G of memory on 4 cores. Note though that this footprint could have been substantially reduced by subsetting all BAM files and the merged assembly to only the Y chromosome, but this would require extra processing and was not required for analysis in Ballgown.

4.2.3 Performance on Clinical Datasets

Next, we carried out experiments designed to represent realistic differential expression scenarios: usually some, but not all, transcripts are truly differentially expressed between populations. We evaluated differential expression results from Ballgown and Cuffdiff2 (versions 2.0.2 and 2.2.1) on two publicly-available clinical datasets. The first experiment[80] compared lung adenocarcinoma ($n = 12$) and normal control samples ($n = 12$) in nonsmoking female patients. The second experiment[81] compared cells at five developmental stages; we analyzed the data from just two stages: embryonic stem cells ($n = 34$) and pre-implantation blastomeres ($n = 78$) – Cuffdiff is only suitable for two-class comparisons. We downloaded the Cuffdiff 2.0.2 output from both studies from InSilico DB[82] on March 5th, 2014. From this output, we extracted isoform-level p-values for cancer/normal and cell type comparisons. We also reformatted the available Cuffdiff 2.0.2 FPKM values and applied Ballgown's linear models to test differential expression. The parameters for the software used by *InSilico DB* were *cufflinks, cuffmerge, cuffdiff: v 2.0.2*, *cufflinks -p 6 -q*, *tophat: v 2.0.4 –mate_inner_dist 80 –no-coverage-search* (personal communication Alain Coletta, from InSilico DB).

We also wanted to run the latest versions of TopHat, Cufflinks, and Cuffdiff, so we downloaded the raw sequencing reads from both experiments from the NCBI Sequence Read Archive.[83] The analysis steps were the same as the steps outlined for processing

the GEUVADIS dataset in Section 5.3, except Tophat version 2.0.11 and Cufflinks version 2.2.1 were used. In addition, there was a small change at the Cufflinks step: because the data sets in InSilico DB were created by estimating transcript abundances for pre-annotated isoforms, we did the same when we processed the data ourselves. This means we ran Cufflinks with the `-G` option and estimated FPKM values for Illumina's iGenomes annotated genes for hg19. These are the isoforms considered in the analysis results. We compared the differential expression results from Cuffdiff (versions 2.0.2 and 2.2.1), the linear models from Ballgown, the empirical Bayes linear modeling framework implemented in limma,[46] and EBSeq,[70] a Bayesian framework designed for isoform-level differential expression.

On these datasets, the p-value distributions from the linear model tests implemented in Ballgown were reasonable, as were the p-value distributions from Cuffdiff version 2.2.1, though Cuffdiff 2.2.1 was more conservative than Ballgown. Results from Cuffdiff version 2.0.2 showed noticeable conservative bias (Figure 4.6).

More specifically: we know that transcript-level differential expression analysis comparing lung adenocarcinoma to normal cells, or comparing embryonic stem cells to pre-implantation blastomeres, should show a strong differential expression signal, especially considering the sample sizes for these experiments. In the cancer vs. normal comparison, there were 19,748 transcripts with average FPKM greater than 1. Cuffdiff (version 2.2.1) identified 4608 of these transcripts as differentially expressed ($q < 0.05$). F-tests comparing nested linear models, as implemented in Ballgown (Section

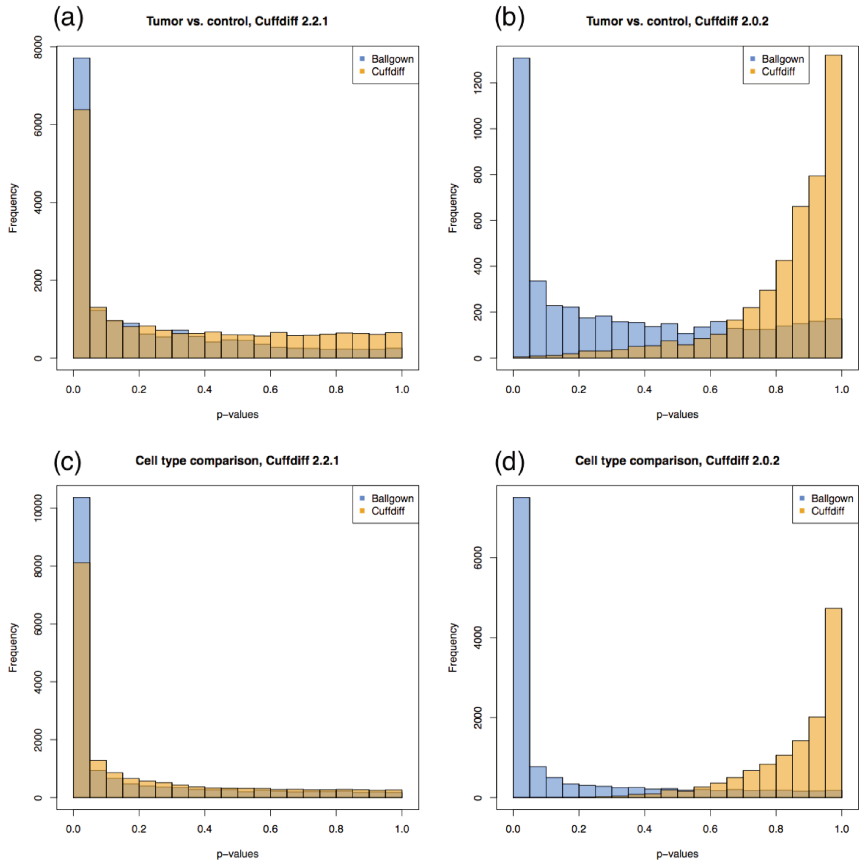

Figure 4.6: Comparison of statistical significance estimates between Cuffdiff and linear models in real datasets a. Histograms of p-values from a comparison of 12 lung adenocarcinomas and 12 normal controls from female patients who never smoked. Ballgown in blue, Cuffdiff (2.2.1) in orange. **b.** Same comparison as in panel (a), but using the Cuffdiff version 2.0.2 results available from InSilico DB. Cuffdiff version 2.0.2 had a strong conservative bias. Linear model results from Ballgown differ from panel (a) because the FPKM estimates used were from an older version of Cufflinks, though the linear model results do not demonstrate conservative bias. **c.** Histograms of p-values from the comparison of 78 pre-implantation blastomere samples and 34 embryonic stem cell samples (Ballgown in blue, Cuffdiff (2.2.1) in orange). **d.** Same comparison as in panel (c), but using the Cuffdiff version 2.0.2 results available from InSilico DB. As in panel (b), Cuffdiff 2.0.2 showed a strong conservative bias.

5.2.3), flagged 8875 of these highly-expressed transcripts as differentially expressed. Of 27,058 transcripts tested, EBSeq called 8736 differentially expressed (posterior probability of differential expression of at least 0.95). Similarly, in the cell type dataset, there were 16,430 transcripts with mean FPKM greater than 1. Cuffdiff (2.2.1) calls 6816 of these differentially expressed ($q < 0.05$) while Ballgown calls 9701 of them differentially expressed. And of 15,462 transcripts tested, EBSeq identifies 10,307 with posterior probabilities of differential expression of at least 0.95.

While both linear modeling and Cuffdiff produced reasonable p-value distributions for these experiments (Figure 4.6a,c), the relative numbers of differentially expressed transcripts discovered and the p-value distribution shapes show that Cuffdiff is more conservative than the linear models. On its own, this result does not necessarily mean that Cuffdiff (2.2.1) is too conservative, but Cuffdiff also produced conservative p-value distributions in the negative and positive control experiments (Section 4.2), we have prior knowledge that the differential expression signal should be quite strong in a tumor/normal or a cell type comparison, and the numbers of discoveries made by another published differential expression method (EBSeq) align more closely with the results from the linear model comparisons. Together these results indicate that conservative bias persists in Cuffdiff (2.2.1). Past versions of Cuffdiff, particularly 2.0.2, produced extremely conservative results on these datasets (Figure 4.6b,d), calling just 1 of 4454 highly-expressed transcripts in the tumor/normal dataset differentially expressed, while Ballgown's linear models identified 774, using

the same FPKM estimates. Similarly, in the cell type dataset, Cuffdiff 2.0.2 found 0 of 12,469 highly-expressed transcripts to be differentially expressed ($q < 0.05$) between embryonic stem cells and preimplantation blastomeres, while the linear model tests in Ballgown found 6964. These results in large-scale studies suggest that Cuffdiff's statistical significance estimates were strongly conservatively biased in version 2.0.2. While version 2.2.1 is better, Cuffdiff is still not producing uniformly-distributed null p-values and is more conservative than other differential expression methods.

4.2.4 Simulation Study

We also carried out two simulation studies using data simulated with the Polyester package (Chapter 6) that demonstrated improved sensitivity and specificity estimates for Ballgown compared with Cuffdiff. Detailed simulation methods are described in Chapter 5.4. Briefly, two scenarios were simulated. In both scenarios, a two-group experiment with 10 biological replicates in each group was generated. In the first scenario, differential expression was set at the FPKM level – i.e., a 2x fold change between groups for a transcript was indicated by a doubling the FPKM value in one of the groups for that transcript. In the second scenario, differential expression was directly set at the read level – i.e., a 2x fold change between groups for a transcript was indicated by doubling the mean number of reads generated from that transcript.

The results for the first simulation, where the differential expression was set at the FPKM level, were that Cuffdiff (2.2.1) showed the same conservative bias we observed

in the negative control experiment and possibly in the InSilico DB experiments. Using the q-value as a significance cutoff, Cuffdiff called 1 transcript differentially expressed (controlling FDR at the 5% level), compared to 56 using Ballgown's F-tests for nested linear models (Section 5.2.3). Accordingly, the p-value distributions showed similar patterns to those we observed in the adenocarcinoma and developmental cell datasets (Figure 4.7a). While the accuracy of the transcript rankings was comparable for both methods – for the linear models in Ballgown, 81 of the top 100 transcripts called differentially expressed were truly differentially expressed for Ballgown versus 85 for Cuffdiff 2.2.1 – an ROC curve based on q-value cutoff shows Ballgown outperforming Cuffdiff in terms of sensitivity and specificity (Figure 4.7), and the significance cutoffs for Ballgown are much more reasonable.

We hypothesize that transcript length normalization may have something to do with the problems observed in Cuffdiff's statistical significance estimation, because in the second simulation scenario, where the number of reads sampled from each transcript was independent of its length, Cuffdiff performed comparably to the linear model framework included in Ballgown, and both seemed to be performing accurately. The p-value histograms for both methods showed uniformly-distributed p-values at the high end and some signal at the low end, as expected (Figure 4.7c), and the ROC curves are approximately equivalent; both display high sensitivity and specificity (Figure 4.7d). In this scenario, of the top 100 transcripts ranked by each method, 96 are truly differentially expressed for Cuffdiff and 91 are for the linear models

implemented in Ballgown. This shows that the models implemented in Cuffdiff are accurate under some conditions – e.g., when the number of sequencing reads from a transcript is unrelated to its length – that may be somewhat unrealistic.

4.3 Analyzing RNA-seq experiments with Complex Designs

One advantage of the Ballgown framework over Cuffdiff and EBSeq is the option to compare any nested set of models to test for differential expression, or to apply standard differential expression tools in Bioconductor, such as the limma package.[46] To demonstrate the flexiblity of linear models like those in Ballgown or limma, we carried out two popular analyses that have not been possible with standard transcriptome assembly and differential expression tools: (1) modeling expression as a smooth function of a continuous covariate and (2) an eQTL analysis.

4.3.1 Effect of RIN on Transcript Expression

In the first analysis, we treated RNA Integrity Number (RIN)[84] as a continuous covariate[85] and used Ballgown's modeling framework to discover transcripts in the GEUVADIS dataset[77,78] (Section 5.3) whose expression levels were significantly associated with RIN. Detailed methods for this analysis are described in Section 5.5. Of

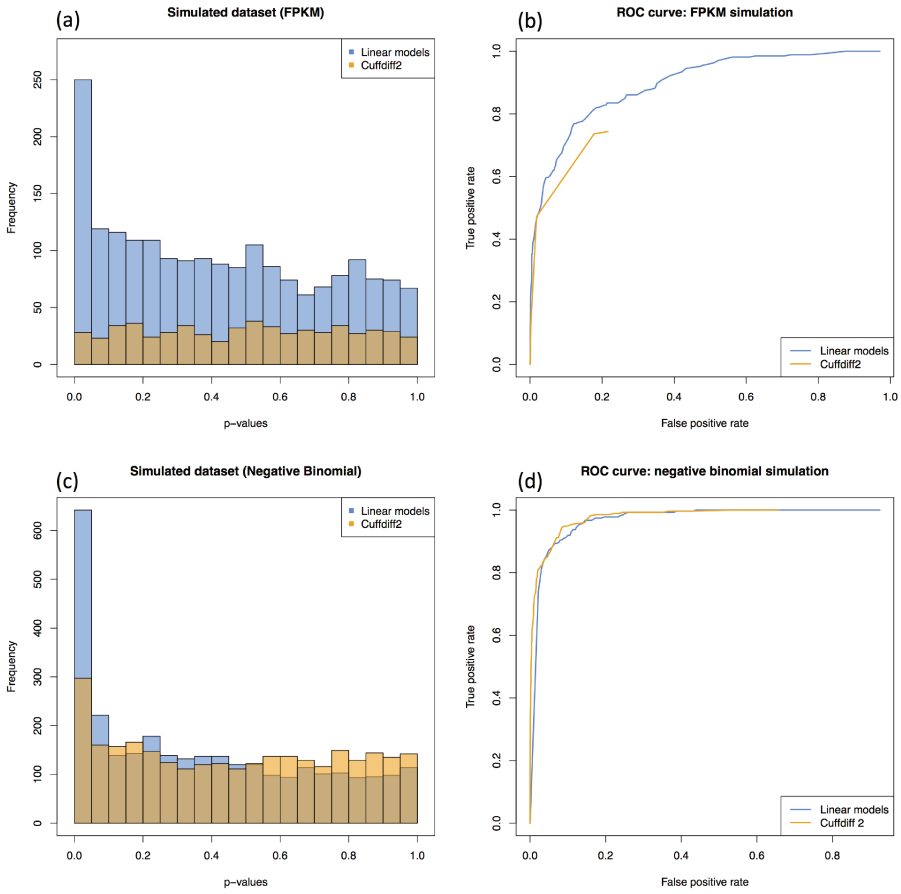

Figure 4.7: Comparison of statistical significance between Cuffdiff and linear models in Ballgown in simulated datasets a. Histograms of p-values from a simulated data set of 2,745 transcripts where differential expression was induced between 10 cases and 10 controls in 10% of transcripts at the FPKM level (Ballgown in blue, Cuffdiff in orange). **b.** ROC curve comparing the abilities of Cuffdiff and linear modeling to identify differentially expressed transcripts in the FPKM simulation based on q-value. **c.** Histograms of p-values from a simulated data set of 2,745 transcripts in 10 cases and 10 controls, where 10% of transcripts were simulated to be differentially expressed, but the number of reads generated from each transcript was independent of transcript length. **d.** ROC curve comparing the abilities of Cuffdiff and linear modeling to identify differentially expressed transcripts in the transcript-length-independent simulation study.

43,622 assembled transcripts with average FPKM above 0.1, 19,203 showed a significant effect ($q < 0.05$) of RIN on expression, using a natural cubic spline model for RIN and adjusting for population and library size.[86]

A previous analysis of the GEUVADIS data modeled variation in RNA-quality as a linear effect.[78] We fit a model with a linear RIN effect and population and library size adjustments to each transcript and identified an enrichment of transcripts with a positive correlation between FPKM values and RNA-quality (Figure 5.1). To investigate the impact of using a more flexible statistical model to detect RIN artifacts, we tested whether a cubic polynomial fit for RIN on transcript expression was significantly better than simply including a linear term for RIN after adjusting for population. We compared the cubic and linear fits on 43,622 transcripts with average FPKM greater than 0.1 across all samples. We found that the cubic fit was significantly better than the linear fit ($q < 0.05$) for 1,499 of the 43,622 transcripts (Figure 4.8), suggesting that flexible non-linear models may be helpful when measuring the relationship between quantitative covariates and transcript abundance levels.

4.3.2 Transcript-level eQTL Analysis

To further illustrate the flexibility of using the post-processed Ballgown data for differential expression analysis, we next carried out an eQTL analysis of the 464 non-duplicated GEUVADIS samples (as indicated by the initial investigators[77]), across all populations. We used the processed transcriptome data described in Section 5.3

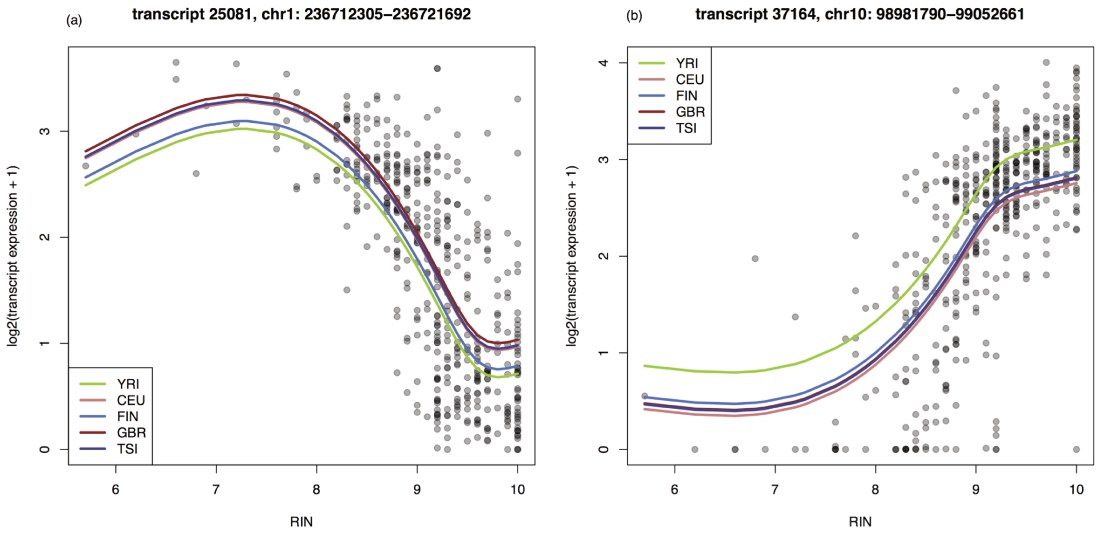

Figure 4.8: Non-linear effects of RNA quality on transcript expression. These two transcripts (FDR < 0.001) and 1,497 others showed a relationship with RNA quality (RIN) that was significantly better captured by a non-linear trend with three degrees of freedom than a standard linear model. Colored lines shown are predicted values from a natural cubic spline fit and represent predictions for the specified population, assuming average library size.

as the expression data set. In addition, we downloaded the GEUVADIS genotype data from `ftp://ftp.ebi.ac.uk/pub/databases/microarray/data/experiment/GEUV/E-GEUV-1/genotypes/`. We filtered to only SNPs with a minor allele frequency greater than 5% and to transcripts with average FPKM across replicates of more than 0.1. This filtering left us with 7,072,917 SNPs and 44,140 transcripts in the analysis. We further constrained our analysis to search for cis-eQTLs, where the genotype and transcript pairs were within 1000kb of each other, which resulted in 218,360,149 SNP-transcript pairs in total. We took the log2 transform of the transcript-level FPKM values and used the MatrixEQTL package[87] to perform the eQTL analysis, which tested an additive linear regression model for the SNPs. We adjusted for the first three principal components of the genotype data,[88] calculated using the Plink software,[89] and the first three principal components of the observed transcript FPKM data[90] in the eQTL model fits.

We recorded the histogram of p-values from all transcript-SNP assoiation tests (Figure 4.9a), and from those p-values, we estimated the fraction of null hypotheses based on the distribution of observed p-values[52] to be $\hat{\pi_0} = 0.942$. The p-value histogram and QQ-plot of -log10(p-values) (Figure 4.9b) versus their theoretical distribution under the null do not show any gross deviations suggesting unmodeled confounding.[91]

To determine overlaps between transcripts we called significant and transcripts previously called significant,[78] we downloaded the list of previously-found significant

cis-eQTL from `ftp://ftp.ebi.ac.uk/pub/databases/microarray/data/experiment/GEUV/E-GEUV-1/genotypes/` for the EUR and YRI populations. We identified all Ensembl genes overlapped to any degree by each assembled transcript. We then calculated the number of gene-SNP pairs in common between the GEUVADIS EUR and YRI analyses and our eQTL analysis.

We identified significant eQTL at the FDR 1% level for 17,276 transcripts overlapping 10,524 unique Ensembl-annotated genes. Our global estimate of the number of the percentage of null hypotheses ($\hat{\pi_0} = 0.942$) indicates that 5.8% of SNP-transcript pairs show differential expression. 57% and 78% of the transcript-SNP pairs called significant in the original analysis of the EUR and YRI populations,[78] respectively, appeared in our list of significant transcript eQTL. 14% of eQTL pairs were identified for transcripts that did not overlap Ensembl annotated transcripts: an example is shown in Figure 4.10. The eQTL analysis was performed in 2 hours and 3 minutes on a standard Desktop computer.

4.4 Ballgown as a Tool for Exploring Alternatives to FPKM

Ballgown offers researchers the flexibility to explore the effects of using alternative expression measurements for analysis. There are two major classes of statistical methods for differential expression analysis of RNA-seq: those based on RPKMs or FPKMs, as

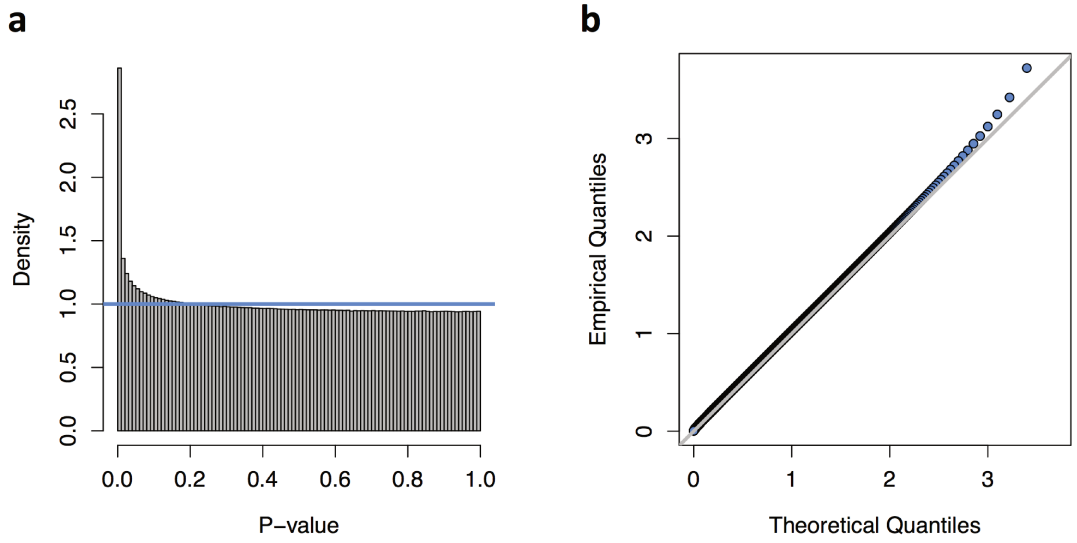

Figure 4.9: Distribution of statistical significance scores for all cis-eQTL tests a. P-value histogram for all p-values from cis-eQTL tests, the estimated fraction of null hypotheses is 94.2%. **b.** QQ-plot of -log10(p-values) versus theoretical quantiles shows no gross deviation from expected behavior.

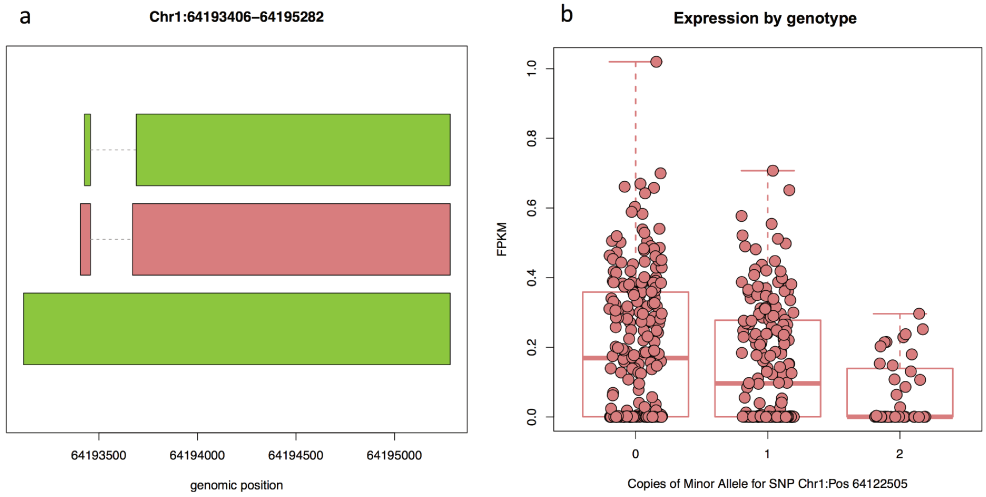

Figure 4.10: Example of an assembled transcript in the GEUVADIS that does not overlap any annotated transcripts, but shows a significant eQTL. Panel (a) displays transcript structures for the locus in question; Panel (b) is a boxplot of the FPKM transcripts for the middle (red) transcript from panel (a), which shows a consistent and statistically significant eQTL.

exemplified by Cufflinks, and those based on counting the reads overlapping specific regions, as exemplified by DESeq[29] and edgeR.[27] Tablemaker outputs both FPKM estimates from Cufflinks and average coverage of each exon, intron, and transcript. The `rsem-calculate-expression` utility outputs TPM (transcripts per million)[92] as the normalized transcript expression measurement. Here we only compare FPKM and average coverage, but note that comparisons with TPM are now possible with the Ballgown software. In this particular experiment, we investigated the effect of expression measurement using both simulated data and the GEUVADIS dataset, and we confirmed that, as expected, differential expression results using average coverage and using FPKM were strongly correlated.

More specifically, we compared previous differential expression results (Sections 4.3.1 and 5.4), which were based on measuring transcript abundance using FPKM, with analyses using average per-base read coverage as the transcript expression measurement instead. Doing this comparison was straightforward, since several different expression metrics are available in the Ballgown objects created for the previous analyses. First, we used our simulated dataset (Section 5.4) to investigate the impact of using average coverage as the transcript expression measurement instead of FPKM, as was done in our previous analyses. To do this comparison, we re-analyzed the data we simulated in the scenario where differential expression occurred at the FPKM level (Figure 4.7a-b) but used average coverage as the transcript-level expression measurement. The differential expression rankings measuring transcript expression with

FPKM and with average coverage were highly correlated (Figure 4.11a), $r = 0.66$. The p-value distribution using average coverage (Figure 4.11b) was similar to the p-value distribution using FPKM (Figure 4.7a), though only 25 transcripts were found to be differentially expressed ($q < 0.05$), compared to 56 using FPKM. We also observed correlated ranks ($r = 0.57$) between the differential expression results testing whether RIN value affected expression in the GEUVADIS dataset (Figure 4.11c). These results suggest that coverage – an expression measurement that is much easier to estimate than FPKM – is potentially a viable alternative expression metric for use in isoform-level differential expression analyses. Ballgown allows users to perform analyses with whatever expression measurements are available for their transcriptome, so for example, RSEM[67] users can use such as transcripts per million (TPM)[92,93] as an expression measurement. The framework also facilitates easy exploration of the different measurement options.

4.5 Computational Efficiency

The linear model differential expression testing framework built into Ballgown or limma provides computational benefits over Cuffdiff and EBSeq. TopHat and Cufflinks can be run on each sample separately, but Cuffdiff must be run on all samples simultaneously. While Cuffdiff can make use of many cores on a single computer, is not parallelizable across computers. It has been noted that Cuffdiff can take weeks

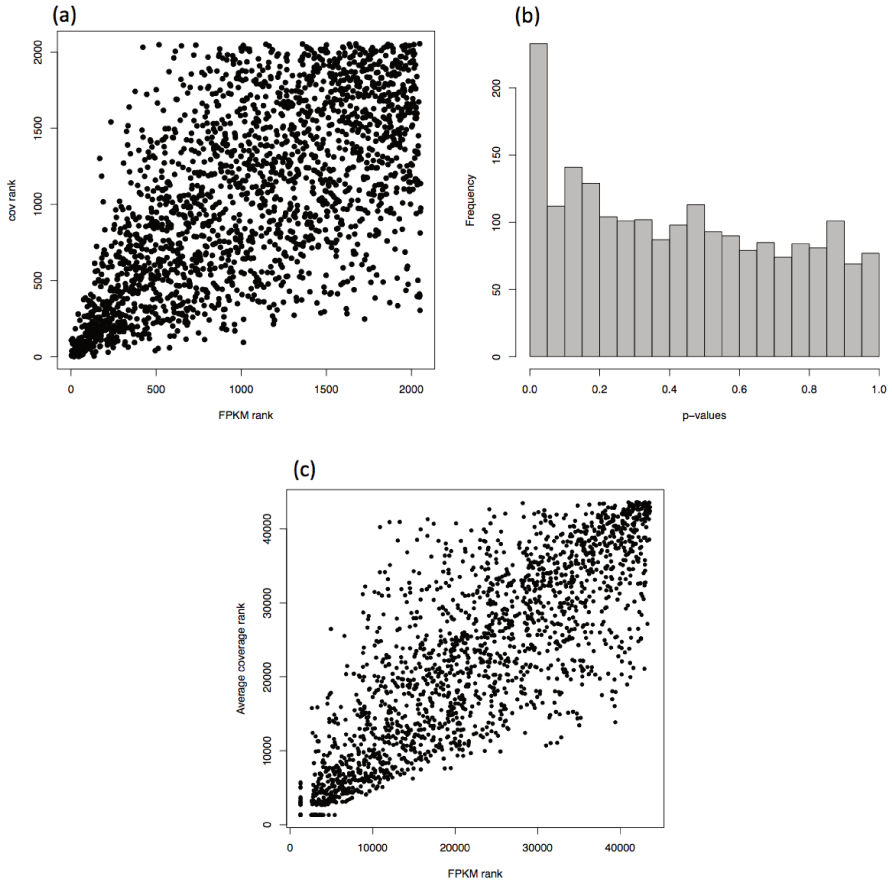

Figure 4.11: Using average per-base coverage as transcript expression measurement instead of FPKM. a. Differential expression ranks for transcripts in a case/control simulation ($n = 10$ per group), using FPKM as the expression measurement (x-axis) vs. using average coverage (y-axis). **b.** Distribution of p-values from differential expression tests between the 10 cases and 10 controls, using average coverage as the expression measurement. This distribution is very similar to the distribution observed when using FPKM as the expression measurement (Figure 4.7a). **c.** Rankings of the effect of RIN on transcript expression in the GEUVADIS dataset, using FPKM as the transcript expression measurement (x-axis) vs. using average coverage (y-axis). For visibility, 2000 transcripts were randomly sampled from the dataset for the plot.

or longer to run on experiments with a few hundred samples. This issue has led consortia and other groups to rely on unpublished software for transcript abundance estimation.[65,77] In addition, EBSeq took 6909 seconds (1.9 hours) to perform a differential expression analysis for the positive control experiment (Section 4.2.2), while Ballgown ran in less than 1 second, and Cuffdiff took 58 hours. EBSeq also took 6.1 hours to analyze the tumor/normal InSilico DB dataset from Section 4.2.3 (after abundance estimation), while Cuffdiff 2.2.1 took 9 hours 46 minutes (including abundance estimation) and Ballgown's statistical models ran in 0.7 seconds (after abundance estimation). Overall, these times indicate that Ballgown is markedly faster than EBSeq, and very likely much faster than Cuffdiff: The Ballgown result does not include Tablemaker time, but we note that Tablemaker is parallelizable across samples and is usually much shorter than Cuffdiff time, e.g., Figure 4.12. Also, Cuffdiff scales quite poorly as the number of biological replicates increases.

We compared each component of the pipeline in terms of computational time on one of our simulated datasets (Section 5.4; the second, simpler scenario) with 20 samples and 2,745 transcripts. The TopHat-Cufflinks-Tablemaker-Ballgown pipeline was fastest, taking about 5.4 minutes per sample for Tablemaker, 2.3 seconds to load transcript data into R and less than 0.1 seconds for differential expression analysis. This is faster than the recently published TopHat-Cufflinks Cuffquant-Cuffdiff pipeline,[94] which required about 3 minutes per sample for Cuffquant and 19 minutes for differential expression analysis with Cuffdiff. The Tablemaker-Ballgown pipeline was also

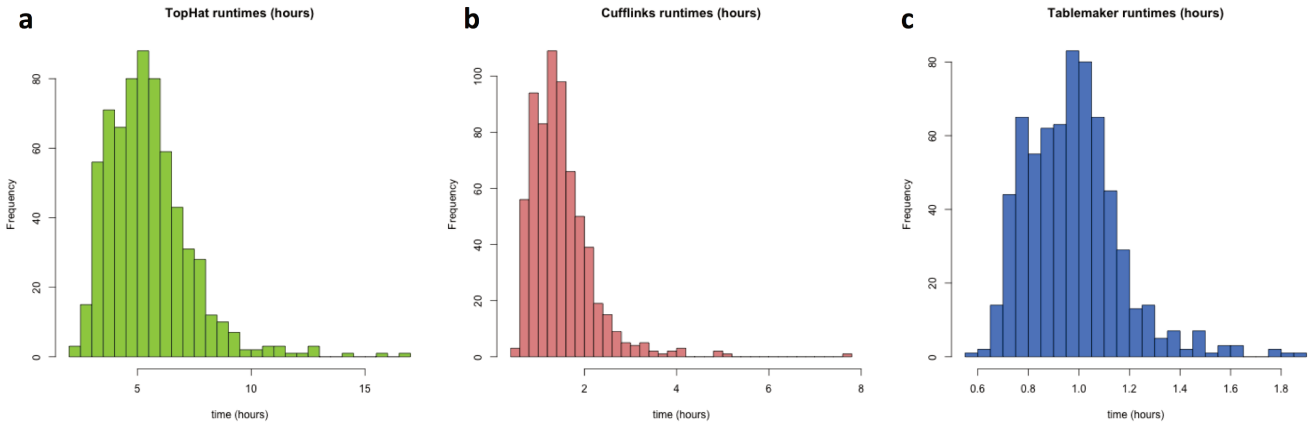

Figure 4.12: Timing results for the 667 GEUVADIS samples at each stage of the pipeline. a. Timing (in hours) for each sample to run through *TopHat2*. **b.** Timing (in hours) for each sample to run through *Cufflinks*. **c.** Timing (in hours) for each sample to run through *Tablemaker*.

substantially faster than directly running Cufflinks-Cuffdiff, where the Cuffdiff step took about 68 minutes. For all these pipelines, TopHat took about 1 hour per sample and Cufflinks about 2 minutes per sample. All possible multicore processes (TopHat, Cufflinks, Cuffdiff, Cuffquant, Tablemaker) were run on 4 cores.

We also calculated the per-sample distribution of processing times for each step in the TopHat-Cufflinks-Tablemaker pipeline for all 667 samples in the GEUVADIS study[77,78] (Figure 4.12). Tablemaker took a median of 0.97 hours per sample (IQR 0.24 hours) on a standard 4 core computer; this calculation can be parallelized across samples. By contrast, Cuffdiff would take months to perform this analysis on a standard 4 core computer. Ballgown multiclass differential expression analysis between the CEU ($n = 162$), YRI ($n = 163$), FIN ($n = 114$), GBR ($n = 115$) and TSI ($n = 93$) samples for 334,206 transcripts took 42 minutes on a single core desktop computer.

These timing results suggest that Ballgown is less computationally intensive than either Cuffdiff or EBSeq, on top of the additional flexibility and accuracy advantages detailed above. Ballgown reduces the computational burden of differential expression analysis of assembled transcriptomes without paying a price in terms of accuracy.

4.6 Summary

Ballgown is designed to be a bridge between upstream assembly tools such as Cufflinks and downstream statistical modeling tools in Bioconductor. The Ballgown suite includes functions for interactive exploration of the transcriptome assembly, visualization of transcript structures and feature-specific abundances for each locus, and post-hoc annotation of assembled features to annotated features. Direct availability of feature-by-sample expression tables makes it easy to apply either the built-in or previously existing differential expression tests or to evaluate other statistical properties of the assembly, such as dispersion of expression values across replicates or genes. The Tablemaker preprocessor writes the tables directly to disk, and they can be loaded into R with a single function call. Together, these software packages provide valuable infrastructure for isoform-level differential expression analysis based on data-driven assemblies.

4.7 Software

1. *Ballgown* - Available from Bioconductor:[25] `http://www.bioconductor.org/packages/release/bioc/html/ballgown.html` Installation instructions and tutorial for use are available in the package vignette, and at `https://github.com/alyssafrazee/ballgown`

2. *Tablemaker* - Available from Figshare. Linux binary: `http://figshare.com/articles/Tablemaker_Linux_Binary/1053137`; Mac OS X binary: `http://figshare.com/articles/Tablemaker_OS_X_Binary/1053136`; source code/installation instructions: `https://github.com/alyssafrazee/tablemaker`

3. *Code for analysis in this chapter* - Available from GitHub: `https://github.com/alyssafrazee/ballgown_code/`

4. *Polyester* - used in section 4.2.4, full description in Chapter 6, software available from Bioconductor: `http://www.bioconductor.org/packages/release/bioc/html/polyester.html`

4.8 Acknowledgements

For this project, JTL, GP, and BL were partially supported by NIH R01 GM105705, and AF was supported by a Hopkins Sommer Scholarship. AEJ was supported by

the Lieber Institute for Brain Development. The authors thank Peter A.C. 't Hoen and Tuuli Lappalainen for providing GEUVADIS quality control data and assistance with contacting ArrayExpress to deposit our processed data. We also acknowledge Cole Trapnell for helpful conversations about Cufflinks architecture and results.

Chapter 5

Supplementary Material: Bridging the gap between transcriptome assembly and expression analysis

This chapter does not stand on its own, but provides supplementary material for Chapter 4. This chapter was published as supplementary material to the published form of Chapter 4 in the journal *Nature Biotechnology*, with contributions from co-authors Geo Pertea, Andrew E. Jaffe, Ben Langmead, Steven L. Salzberg, and Jeffrey T. Leek.

5.1 Tablemaker output files

Tablemaker outputs the following set of related tab-delimited text files. Tablemaker is designed to be run on the output of Cufflinks and Cuffmerge, but Ballgown can be used with any assembly output that can be converted into the following sets of tab-delimited files. In particular, StringTie[66] can be run in `-B` mode to output these files, and Ballgown's data-loading function automatically reads RSEM[67] output.

- *e_data.ctab*: exon-level expression measurements. One row per exon. Columns are *e_id* (numeric exon id), *chr*, *strand*, *start*, *end* (genomic location of the exon), and the following expression measurements for each sample:
  - *rcount*: reads overlapping the exon
  - *ucount*: uniquely mapped reads overlapping the exon
  - *mrcount*: multi-map-corrected number of reads overlapping the exon
  - *cov*: average per-base read coverage
  - *cov_sd*: standard deviation of per-base read coverage
  - *mcov*: multi-map-corrected average per-base read coverage
  - *mcov_sd*: standard deviation of multi-map-corrected per-base coverage
- *i_data.ctab*: intron- (i.e., junction-) level expression measurements. One row per intron. Columns are *i_id* (numeric intron id), *chr*, *strand*, *start*, *end* (genomic

location of the intron), and the following expression measurements for each sample:

  - *rcount*: number of reads supporting the intron
  - *ucount*: number of uniquely mapped reads supporting the intron
  - *mrcount*: multi-map-corrected number of reads supporting the intron

- *t_data.ctab*: transcript-level expression measurements. One row per transcript. Columns are:

  - *t_id*: numeric transcript id
  - *chr*, *strand*, *start*, *end*: genomic location of the transcript
  - *t_name*: Cufflinks-generated transcript id
  - *num_exons*: number of exons comprising the transcript
  - *length*: transcript length, including both exons and introns
  - *gene_id*: gene the transcript belongs to
  - *gene_name*: HUGO gene name for the transcript, if known. This field is empty unless Cufflinks or Cuffmerge was run in annotation mode (with the `-g` flag).
  - *cov*: per-base coverage for the transcript (available for each sample)
  - *FPKM*: Cufflinks-estimated FPKM for the transcript (available for each sample)

- *e2t.ctab*: table with two columns, *e_id* and *t_id*, denoting which exons belong to which transcripts. These ids match the ids in the *e_data* and *t_data* tables.

- *i2t.ctab*: table with two columns, *i_id* and *t_id*, denoting which introns belong to which transcripts. These ids match the ids in the *i_data* and *t_data* tables.

5.2 Data, notation, and statistical models

There are two distinct components to the data that Ballgown is equipped to analyze: (1) the actual structure of the assembled transcriptome (genomic locations of features and the relationships between exons, introns, transcripts and genes) and (2) the expression measurements for the features in the transcriptome. Here we precisely define both the assembly structure and the associated data.

5.2.1 Assembly structure

The transcriptome is assembled based on a set R of aligned RNA-seq reads. We denote the yth read from the zth sample with r_{yz}, where $y = 1, ..., N_z$ and $z = 1, ..., n$, so there are n samples in the study, and sample z has N_z aligned reads.

The transcriptome assembled from the reads consists of four types of features: transcripts, genes, exons, and introns. These features all have start and finishing positions on the genome, which represent using the functions $s()$ and $f()$, e.g., $s(x)$ represents the start position of feature x. The K assembled transcripts are denoted

by t_k, where $k = 1, ..., K$. These transcripts can be organized into G genes, denoted by g_l, $l = 1, ..., G$. Each gene can be represented by a set of transcripts falling within its boundaries:

$$g_l = \{t_k : s(t_k) > s(g_l) \text{ and } f(t_k) < f(g_l)\}$$

The assembly also contains M exons, each of which we represent as a closed interval of genomic locations:

$$e_m = [s(e_m), f(e_m)], m = 1, ..., M$$

With this notation, we can then represent transcript k as a subset of the M exons comprising the assembly:

$$t_k = \{e_m : m \in I_k\}, I_k \subset \{1, ..., M\}$$

Here, I_k is the set of indices of the exons that make up transcript k. Note that the exon e_m can belong to several different transcripts, so the I_ks are not necessarily disjoint. We can then easily define $s(t_k)$ and $f(t_k)$ in terms of exon boundaries:

$$s(t_k) = \min\{s(e_m) : m \in I_k\}$$

$$f(t_k) = \max\{f(e_m) : m \in I_k\}$$

Finally, let w_k represent the wth element of I_k. Then we can denote the wth intron

in transcript k with an open interval:

$$i_{kw} = (f(e_{w_k}), s(e_{(w+1)_k}))$$

In other words, i_{kw} is simply the genomic interval between the wth and $w+1$th exons of transcript k.

With these definitions in place, we can now precisely define the reads r_{yz}. An RNA-seq read is simply a subsequence of an RNA transcript. Using set notation, we can define each read using the form:

$$r_{yz} = \left\{ x \in [E, E'] : E < E' \text{ and } x, E, E' \in \bigcup_{m \in I_k} e_m \text{ for some } k \right\}$$

An assembly algorithm applied to the set of reads r_{yz} produces estimates of the exons: $\hat{e}_m, m = 1, \ldots, M$, transcripts: $\hat{t}_k, k = 1, \ldots K$ of the transcripts and genes: $\hat{g}_l, l = 1, \ldots, G$. Most current statistical models treat this assembly as fixed and correct when performing analyses. But as we will demonstrate in the methods section, assembled transcripts are subject to error and may be improved through statistical analysis.[24,67]

5.2.2 Expression data

Next we can define expression measurements for each type of feature given a particular assembled set of transcripts. Here we define sensible expression measurements that are currently implemented in the Ballgown package, but the statistical models are flexible enough to handle other types of measures as well.

For each sample z, each transcript $\hat{t}_k$ has two measurements that are calculated by our upstream Ballgown preprocessing software: average per-base read coverage: $cov(t_k, z)$ and FPKM (fragments per kilobase of transcript per million mapped reads): $FPKM(t_k, z)$. Currently, these transcript-level measurements are estimated in Cufflinks via maximum likelihood; the procedure has previously been described in detail.[22]

Each gene g_l has one expression measurement for each sample, $FPKM(g_l, z)$. This measurement is reconstructed from the transcripts in g_l as follows: first, the number of fragments per million mapped reads for sample z for each $t_k \in g_l$ is calculated by multiplying $FPKM(t_k, z)$ by the length of transcript t_k in kilobases. The gene's total fragments per million mapped reads is the sum of the transcript-level fragments per million mapped reads for all the transcripts in the gene. Finally, the gene-level FPKM is calculated by dividing the gene's total fragments per million mapped reads by the gene's length.

Tablemaker also calculates average per-base read coverage for each exon in the

assembly, given the assembly structure and the aligned reads R. For sample z, we have:

$$cov(e_m, z) = \frac{\sum_{r_{yz} \in R} \sum_{bp \in [s(e_m), f(e_m)]} \mathbb{1}\{bp \in r_{yz}\}}{f(e_m) - s(e_m) + 1}$$

Each exon also has a raw read count, defined as the number of reads whose alignments overlap that exon:

$$rcount(e_m, z) = \sum_{r_{yz} \in R} \mathbb{1}\{r_{yz} \cap e_m \neq \emptyset\}$$

The main expression measurement for introns is also raw read count, defined as the number of reads whose alignments support the intron in the sense that their alignments are split across that intron's neighboring exons:

$$rcount(i_{kw}, z) = \sum_{r_{yz} \in R} \mathbb{1}\{s(r_{yz}) \in e_m \text{ and } f(r_{yz}) \in e_{m'}\}$$

where $m \leq w_k$ and $m' \geq (w+1)_k$.

5.2.3 Statistical methods for detecting differential expression

Here we outline Ballgown's built-in framework for statistical analysis of transcript and gene abundances. To make the ideas concrete we use FPKM as the expression measurement and transcripts as the feature of interest, but these can be replaced in the following model definitions with any of the expression measurements and any of

the available genomic features in the assembly (genes, transcripts, exons, or introns).

Ballgown's default differential expression tests are implemented as follows: for each transcript $\hat{t}_k$, the following model is fit:

$$h(FPKM(\hat{t}_k, z)) = \alpha_k + \sum_{p=1}^{P} \beta_{pk} X_{zp} + \varepsilon_{zk} \tag{5.1}$$

where:

- $FPKM(\hat{t}_k, z)$ is the FPKM expression measurement for transcript k for sample z
- h is a transformation[40] to reduce the impact of mean-variance relationships observed in the counts.[29] For example, the transformation $h(\cdot) = \log_2(\cdot + 1)$ is commonly applied in the analysis of sequence-count data.[95]
- α_k represents the baseline expression for transcript k
- X_{zp} represents covariate p for sample z. These covariates differ by experiment type. X_{z1} generally represents a library size adjustment for sample z. Assuming c_k represents the 75th percentile of all log FPKM values for transcript k, `ballgown`'s default the covariate X_{1z} is:

$$\sum_k FPKM(\hat{t}_k, z) \mathbb{1}[FPKM(\hat{t}_k, z) <= c_k]$$

This normalization term is derived from the "cumulative sum scaling" (CSS)

normalization approach.[86]

- β_{pk} quantifies the association of covariate p on the expression of transcript k
- ε represents residual measurement error

A flexible approach to differential expression is to compare nested sub models of model (5.1) using parametric F-tests.[46] The null model can be as flexible as any linear contrast of the coefficients β_{pk}, but for simplicity we focus on hypotheses of the form:

$$H_0 : \beta_{pk} = 0, p \in \mathcal{S}$$

versus the alternative:

$$H_a : \beta_{pk} \neq 0 \text{ for some } p \in \mathcal{S}$$

The general principle is that a model including any potential confounders plus the covariate(s) of interest – for example, a 0/1 group indicator for the case/control scenario, several indicator variables for a multi-group comparison, or a generalized additive model[96] for a continuous covariate like time in a timecourse experiment – is compared with a model that includes only the potential confounders. For the two models fit for each transcript k, Ballgown calculates the straightforward statistic

$$F = \frac{\frac{RSS_0 - RSS_1}{P_1 - P_0}}{\frac{RSS_1}{n - P_1}}$$

where RSS_0 represents the residual sum of squares from the model without group

or time covariates, RSS_1 represents the residual sum of squares from the model including the covariates of interest, P_0 is the number of covariates in the smaller model, P_1 is the number of covariates in the larger model, and n is the total number of samples. Under the null hypothesis that the larger model does not fit the data significantly better than the smaller model, this statistic follows an F distribution with $(P_1 - P_0, n - P_1)$ degrees of freedom, so p-values can be generated by comparing the two models for each transcript k.[90] We control for multiple testing using standard FDR controlling procedures.[52]

5.3 Processing the GEUVADIS data

We downloaded the FASTQ files from the GEUVADIS project[77,78] from `http://www.ebi.ac.uk/ena/data/view/ERP001942`. With this data, we:

- Aligned reads with TopHat 2.0.9, using the `-G` option to align reads to the transcriptome first. We used the hg19 genome reference available from the Illumina iGenomes project.
- Assembled sample-specific transcriptomes with Cufflinks 2.1.1, using default options and no annotation
- Merged sample-specific assemblies into an experiment-wide assembly with Cuffmerge 2.1.1

- Estimated feature expression and organized the assembly with Tablemaker so that all files described in Section 5.1 were available.

- Created several Ballgown objects using the Ballgown R package

The resulting Ballgown objects include phenotype data available from several sources, including `http://www.ebi.ac.uk/ena/data/view/ERP001942`, the 1000 Genomes Project,[97] and additional quality control data from GEUVADIS researchers (available at `https://github.com/alyssafrazee/ballgown_code/blob/master/GEUVADIS_preprocessing/GD667.QCstats.masterfile.txt`). The Ballgown R objects are available for download at `http://figshare.com/articles/GEUVADIS_Processed_Data/1130849`. So the objects can be feasibly loaded into memory and stored on disk, a separate object is available for each expression measurement.

5.4 Methods for Simulation Studies

To ensure that the linear models implemented in Ballgown perform accurately, we performed two separate simulation studies. Results are presented in Section 4.2.4. For both studies, reads were generated from 2745 annotated transcripts on Chromosome 22 from Ensembl,[98] using genome build GRCh37 and Ensembl version 74. Data was generated for 20 biological replicates, divided into two groups of 10, where 274 transcripts were randomly chosen to be differentially expressed (at a 6x increase in expression level) in one of the two groups, randomly chosen.

The first simulation study was set up as follows:

- Expression was measured in FPKM. Each transcript's baseline mean FPKM value was determined based on the distribution of mean FPKM values for highly-expressed transcripts in the GEUVADIS dataset. Specifically, the mean of all nonzero FPKM values was calculated for each transcript in the GEUVADIS dataset with mean FPKM larger than 100, and each isoform in the simulated dataset was assigned a randomly selected baseline mean FPKM from this distribution.

- We defined a log-log relationship between a transcript's mean expression level and the variance of its expression levels:

$$\log \text{variance} = 2.23 \log \text{mean} - 3.08$$

  This relationship was estimated empirically from the assembled GEUVADIS transcriptome (transcripts with mean FPKM values greater than 10) using simple linear regression. The GEUVADIS dataset includes both biological and technical replicates, so this model should encompass both biological and technical variability.

- Then, for each transcript, we randomly drew FPKM expression values from a log-normal distribution with the pre-set mean and variance. For the differentially expressed transcripts, the pre-set mean FPKM was 6 times larger in one

group than in the other.

- For each transcript, we also set a sample's expression level to 0 with probability p_0, which was estimated from the GEUVADIS data: for each simulated transcript, p_0 was randomly drawn from the empirical distribution of the proportion of samples with zero expression, over transcripts in the GEUVADIS dataset with mean FPKM larger than 100.

- To translate the pre-set FPKM value into a number of reads to be generated from a transcript for a given sample, we used the definition of FPKM and calculated the number of "fragments" (reads) that should be generated from a transcript by multiplying the set FPKM value by the transcript's length over 1000, then multiplying by an approximate library size of 150,000 reads, over 1 million. The decision to use a mixture of two distributions (log-normal and point mass at 0) was informed by exploratory analysis of the FPKM distributions among several transcripts in the GEUVADIS dataset. The exploratory analysis is available at `http://htmlpreview.github.io/?https://github.com/alyssafrazee/ballgown_code/blob/master/simulations/mean_var_relationship.html`.

This simulation setup made it such that more reads were generated from longer transcripts, as is expected with RNA-seq protocols.

A second simulation was also conducted with a slightly simpler setup:

- Expression was defined directly by the number of reads being generated from each transcript (instead of using FPKM).

- The mean number of reads generated from each transcript was set to be 300, unless the transcript was randomly selected to be overexpressed in one group, in which case, that group's mean read number for that transcript was 1800.

- The actual number of reads to be simulated from a transcript was drawn from a negative binomial distribution with mean $\mu = 300$ or 1800, and size equal to 0.005μ (so, 1.5 for $\mu = 300$ and 9 for $\mu = 1800$). Note that in the negative binomial distribution, the variance is equal to $\mu + \mu^2/\text{size}$.

- Each sample's read counts were scaled and rounded such that approximately 600,000 reads were generated per sample.

For both these scenarios, the specified number of reads was then generated from transcripts using the Polyester Bioconductor package (Chapter 6). These simulated reads were then aligned to the genome using TopHat 2.0.11 (aligning to the annotated transcriptome first with the `-G` option), and the resulting alignments were used to assemble transcripts with Cufflinks 2.2.1. Cuffdiff (2.2.1) was then run on the simulated datasets. For the Ballgown results in Section 4.2.4, we used Tablemaker to organize the output, but because Tablemaker calls Cufflinks version 2.1.1 to estimate per-transcript FPKMs, we updated the `ballgown` object to use the FPKMs written in the `isoforms.read_group_tracking` file by Cuffdiff 2.2.1.

The following models were fit for each transcript in each simulation scenario:

$$H_A \quad : \quad \log_2(FPKM_i + 1) = \beta_0^* + \beta_1^* grp_i + \eta^* q75_i + \epsilon_i^*$$

$$H_0 \quad : \quad \log_2(FPKM_i + 1) = \beta_0 + \eta q75_i + \epsilon_i$$

where grp_i is the value of the group indicator for sample i and $q75$ is a library-size normalizing constant equal to the sum of the log of the nonzero FPKM values to the 75th percentile (known as "cumulative sum scaling" normalization[86]). We then tested the hypothesis $H_0 : \beta_1^* = 0$ versus the alternative that the coefficient was non-zero. For the analysis with average coverage, we replaced $FPKM_i$ with $acov_i$ in the above equations.

We performed simulation studies to precisely assess the accuracy of the differential expression methods. However, assessing the accuracy of transcript-level differential expression is complicated because the annotated transcripts from which reads were generated do not exactly match the assembled transcripts which were tested for differential. This means there is no standard way to define which assembled transcripts should be called differentially expressed. In our accuracy assessments (Figure 4.7), we chose to identify the three closest assembled "neighbors" for each of the 274 truly DE annotated transcripts. Distance was measured by percent overlap, so each annotated transcript's 3 closest assembled neighbors were the 3 transcripts overlapping

it the most. All of these selected "neighbors" were considered as part of the sensitivity and specificity calculations: sensitivity was defined as the ratio of the number of truly differentially expressed annotated transcripts with at least one of its three closest assembled neighbors called differentially expressed to the total number of truly differentially expressed annotated transcripts. Specificity was defined as percentage of "non-neighbor" assembled transcripts that were correctly called not differentially expressed, where "non-neighbor" means the assembled transcript was not one of the three closest to an annotated transcript set to be differentially expressed.

5.5 Methods for Analyzing Effect of RIN on Transcript Expression

To perform the analysis described in Section 4.3.1, we filtered to the 464 unique replicates in the GEUVADIS study[77,78] (Section 5.3) as indicated in the quality control data from the authors, and we analyzed only transcripts with mean FPKM > 0.1 across replicates. We first searched for differential expression with respect to RNA quality (RIN) using the following set of nested linear models to each transcript:

$$H_A \quad : \quad \log_2(FPKM_i + 1) = \beta_0^* + \sum_{t=1}^{4} \beta_t^* \text{spline}_t(RIN_i) + \sum_{p=1}^{5} \gamma_p^* 1(Pop_i = p) + \eta^* q75_i + \epsilon_i^*$$

$$H_0 \quad : \quad \log_2(FPKM_i + 1) = \beta_0 + \sum_{p=1}^{5} \gamma_p 1(Pop_i = p) + \eta q75_i + \epsilon_i$$

Here i indicates sample and the subscript for transcript has been suppressed for clarity. H_0 denotes the null model and H_A denotes the alternative. The first set of terms encode a natural cubic spline fit with 4 degrees of freedom between the RIN values and the FPKM levels; the term $\text{spline}_t(RIN_i)$ refers to the tth B-spline basis term for sample i. The second set of terms encode a factor model for the relationship between population and FPKM and the last term is a library size normalization term that consists of the sum of log of the the non-zero FPKMs up to the 75th percentile for that sample ("cumulative sum scaling" normalization[86]). We then tested the hypothesis that $H_0 : \beta_1 = \beta_2 = \beta_3 = \beta_4 = 0$ versus the alternative that at least one coefficient was non-zero. All transcripts with a q-value[52] less than 0.05 were called significant.

Next we attempted to identify transcripts where expression was significantly better explained by a non-linear polynomial fit than by a linear trend. We fit the following nested set of models:

$$H_A \quad : \quad \log_2(FPKM_i + 1) = \beta_0^* + \sum_{t=1}^{3} \beta_t^* RIN_i^t + \sum_{p=1}^{5} \gamma_p^* 1(Pop_i = p) + \eta^* q75_i + \epsilon_i^* \quad (5.2)$$

$$H_0 \quad : \quad \log_2(FPKM_i + 1) = \beta_0 + \beta_1 RIN_i + \sum_{p=1}^{5} \gamma_p 1(Pop_i = p) + \eta q75_i + \epsilon_i \quad (5.3)$$

and tested the hypothesis that $H_0 : \beta_2 = \beta_3 = 0$ versus the alternative that at least one of the higher order polynomial coefficients was nonzero. Again, all transcripts with a q-value[52] less than 0.05 were called significant.

The transcripts in Figure 4.8 were statistically significant at the FDR 5% level for this second analysis. In Figure 4.8, the curves represent the fitted values for the average library size within each population. We show one example each of a positive and negative relationship between expression and RIN. While there were several examples of associations in both directions, there were more positive associations, as expected (Figure 5.1).

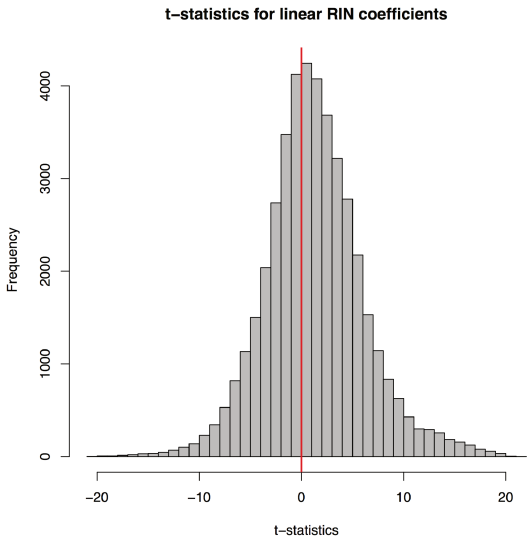

Figure 5.1: Distribution of t-statistics for the linear RIN term for GEUVADIS transcripts. These are moderated t-statistics calculated with *limma* for the β_1 coefficient in model 5.1, indicating directionality of the RIN-FPKM relationship. We observe associations in both directions, but as expected, there are more positive associations.

Chapter 6

Simulating RNA-seq datasets with differential transcript expression

This chapter describes work that at the time of publication of this thesis is under minor revision at the journal *Bioinformatics*, with contributions from co-authors Andrew E. Jaffe, Ben Langmead, and Jeffrey T. Leek.

6.1 Introduction

RNA sequencing (RNA-seq) experiments have become increasingly popular as a means to study gene expression. There are a range of statistical methods for differential expression analysis of RNA-seq data[26] (see also Chapters 2 and 4). Developers of statistical methodology for RNA-seq need to test whether their tools are

performing correctly. Often, accuracy tests cannot be performed on real datasets because true gene expression levels and expression differences between populations are usually unknown, and spike-in experiments are costly in terms of both time and money.

Instead, researchers often use computational simulations to create datasets with a known signal and noise structure. Typically, simulated expression measurements used to evaluate differential expression tools are generated as gene counts from a statistical model like those used in common differential expression tools.[27,29] But these simulated scenarios do not account for variability in expression measurements that arises during upstream steps in RNA-seq data analysis, such as read alignment or read counting. Polyester is a new R package for directly simulating RNA-seq reads. Polyester's main advantage is that users can simulate sequencing reads with specified differential expression signal for either genes or isoforms. This allows for investigation of sources of variability at multiple points in RNA-seq pipelines.

Existing RNA-seq simulators that generate sequencing reads are not designed for simulating experiments with replicates and specified differential expression signal. For example, the `rsem-simulate-reads` utility shipped with RSEM[67] requires a time-consuming first step of aligning real sequencing reads to develop a sequencing model before reads can be simulated, and differential expression simulation is not built-in. Neither Flux Simulator[62] nor BEERS[99] have a built-in mechanism for introducing differential expression. These simulators also do not provide methods for defining

a model for biological variability across replicates or specifying the exact expression level of specific transcripts. TuxSim has been used to simulate RNA-seq datasets with differential expression,[24] but it is not publicly available.

Polyester was created to fulfill the need for a tool to simulate RNA-seq reads for an experiment with replicates and well-defined differential expression. Users can easily simulate small experiments from a few genes or a single chromosome. This can reduce computational time in simulation studies when computationally intensive steps such as read alignment must be performed as part of the simulation. Polyester is open-source, cross-platform, and freely available for download from Bioconductor[25] at `http://www.bioconductor.org/packages/release/bioc/html/polyester.html`.

6.2 Methods

6.2.1 Input

Polyester takes annotated transcript nucleotide sequences as input. These can be provided as cDNA sequences in FASTA format, labeled by transcript. Alternatively, users can simulate from a GTF file denoting exon, transcript, and gene structure paired with full-chromosome DNA sequences. The flexibility of this input makes it possible to design small, manageable simulations by simply passing Polyester a FASTA or GTF file consisting of feature sets of different sizes. Efficient functions for reading,

subsetting, and writing FASTA files are available in the *Biostrings* package,[100] which is a dependency of Polyester.

6.2.2 RNA-seq data as a basis for model parameters

Several components of Polyester, described later in this section, require parameters estimated from RNA-seq data. We have pre-estimated these parameters from public data and included the estimates as defaults in the Polyester R package, so users do not need to spend time and computing power getting the estimates themselves unless they choose to change the defaults. To get our estimates, we analyzed RNA-seq reads from 7 biological replicates in the public GEUVADIS RNA-seq data set.[77,78] The 7 replicates were chosen by randomly selecting one replicate from each of the 7 laboratories that sequenced samples in the study. These replicates represented 7 people from three different HapMap populations: CEU (Utah residents with Northern and Western European ancestry), TSI (Tuscani living in Italy), and YRI (Yoruba living in Ibadan, Nigeria). Data from the GEUVADIS study is available from the ArrayExpress database (www.ebi.ac.uk/arrayexpress) under accession numbers E-GEUV-1 through E-GEUV-6. We specifically used TopHat read alignments for these 7 replicates, under accession number E-GEUV-6. The reads were 75bp, paired-end reads.

Also available for the GEUVADIS data set is a fully processed transcriptome assembly, created based on the RNA-seq reads from all 667 replicates in the GEUVADIS study without using a reference transcriptome. This assembly was built using Cufflinks and processed with the *Ballgown* R package,[101] and is available for direct download as an R object.[102] This processed transcriptome assembly was described in Section 5.3. We use this processed data in several of the following sections.

6.2.3 Expression Models

A key feature of Polyester is that the analyst has full control over the number of reads that are generated from each transcript in the input file, for each replicate in the experiment. Polyester ships with a built-in model for these read numbers, or the model can be explicitly specified by the end user.

6.2.3.1 Built-in negative binomial read count model

The built-in transcript read count model assumes the number of reads to simulate from each transcript is drawn from a negative binomial distribution, across biological replicates. The negative binomial model for read counts has been shown to satisfactorily capture biological and technical variability.[27,29] In Polyester, differential expression between experimental groups is defined by a multiplicative change in the mean of the negative binomial distribution generating the read counts.

Specifically, define Y_{ijk} as the number of reads simulated from replicate i, exper-

imental condition j, and transcript k ($i = 1, ..., n_j$; $j = 1, ..., J$; and $k = 1, ..., N$; where n_j is the number of replicates in condition j, J is the total number of conditions, and N is the total number of transcripts provided). The built-in model in Polyester assumes:

$$Y_{ijk} \sim \text{Negative Binomial}(\text{mean} = \mu_{jk},\ \text{size} = r_{jk})$$

In this negative binomial parameterization, $E(Y_{ijk}) = \mu_{jk}$ and $\text{Var}(Y_{ijk}) = \mu_{jk} + \frac{\mu_{jk}^2}{r_{jk}}$, so each transcript's expression variance across biological replicates is quadratically related to its baseline mean expression. The quantity $\frac{1}{r_{jk}}$ is commonly referred to as the dispersion parameter in this parameterization.[27,103,104] The user can provide μ_{jk} for each transcript k and experimental group j. In particular, the user can relate transcript k's length to μ_{jk}. Also, this flexible parameterization reduces to the Poisson distribution as $r_{jk} \to \infty$. Since the Poisson distribution is suitable for capturing read count variability across technical replicates,[34] users can create experiments with simulated technical replicates only by making all r_{jk} very large. By default, $r_{jk} = \frac{\mu_{jk}}{3}$, which means $\text{Var}(Y_{ijk}) = 4\mu_{jk}$. The user can adjust r_{jk} on a per-transcript basis as needed, to explore different mean/variance expression models.

When $J = 2$, differential expression is set by providing a fold change λ between the two conditions for each transcript. Initially, a baseline mean μ_k is provided for each transcript, and μ_{1k} and μ_{2k} are set to μ_k. Then, if fold change λ is provided, μ_{1k}

and μ_{2k} are adjusted: if $\lambda > 1$, $\mu_{1k} = \lambda\mu_k$, and if $\lambda < 1$, $\mu_{2k} = \frac{1}{\lambda}\mu_k$. The number of reads to generate from each transcript is then drawn from the corresponding negative binomial distribution. When $J > 2$, the count for each transcript, y_{ijk}, is generated from a negative binomial distribution with overall mean μ_k and size r_{jk}. Differential expression can be set using a fold change matrix with N rows and J columns. Each count y_{ijk} is multiplied by entry k, j of the fold change matrix.

6.2.3.2 Options for adjusting read counts

Users can optionally provide multiplicative library size factors for each replicate in their experiment, since the total number of reads (sequencing depth) is usually unequal across replicates in RNA-seq experiments.[13] All counts for a replicate will be multiplied by the library size factor.

GC (guanine-cytosine) content is known to affect expression measurements for genomic features, and the effect varies from sample to sample.[41,42,105] Polyester includes an option to model this GC bias in the simulated reads: for each biological replicate in the simulated data set, the user can choose one of 7 built-in GC content bias models, where one model was estimated from each of the 7 GEUVADIS replicates described in Section 6.2.2. To compute the models, we calculated transcript-level read counts for each replicate based on transcript length, sequencing depth, and the observed FPKM for the transcript. By definition of FPKM, read counts can be directly calculated using these inputs. We then centered the transcript counts around the overall mean

transcript count for that replicate, and modeled the centered counts as a smooth function of the transcript GC content using a loess smoother with span 0.3, analgous to smoothers previously used for modeling GC content.[105]

Transcript GC content was then calculated as the percentage of the annotated hg19 nucleotides falling in the boundaries of the assembled transcript that were G or C. The fitted loess curve defines a function that returns the average deviation from the overall mean transcript count for a transcript with a given GC content percentage. If there is no GC bias, the deviation would be 0. GC bias is added to replicates in Polyester after transcript-level counts have been specified by increasing or decreasing the count by the predicted deviation for that transcript's GC content. The 7 loess curves included in Polyester are shown in Figure 6.1. Users can also provide loess models from their own data as GC bias models if desired.

6.2.3.3 User-defined count models

As an alternative to the built-in negative binomial model, Polyester allows users to individually specify the number of reads to generate from each transcript, for each sample. This gives researchers the flexibility to design their own models for biological and technical variability, simulate complex experimental designs, such as timecourse experiments, and explore the effects of a wide variety of experimental parameters on differential expression results. This transcript-by-sample read count matrix can be created within R and input directly into Polyester's read simulation function.

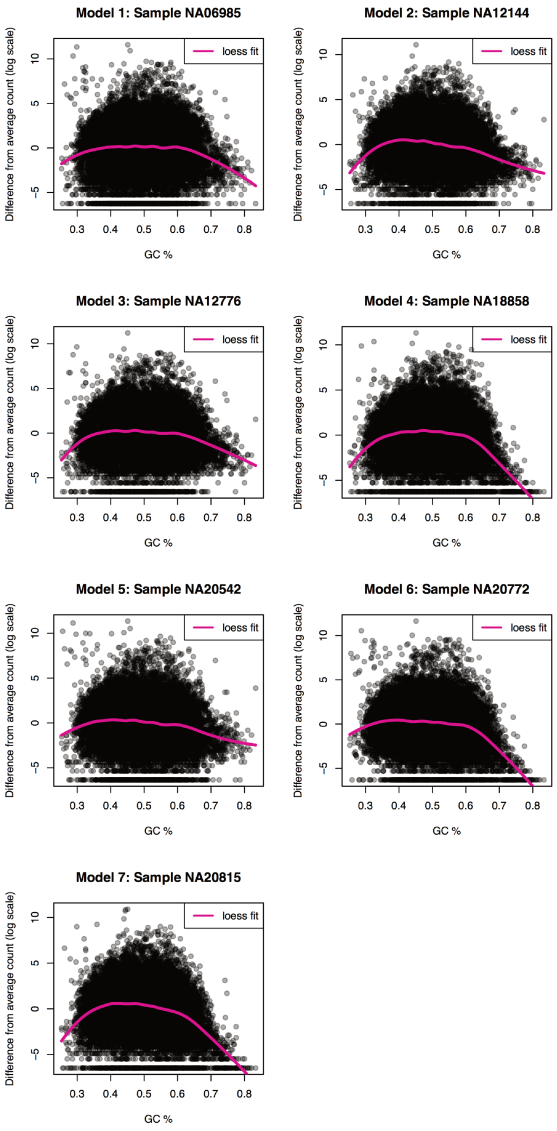

Figure 6.1: GC content models for expression included in Polyester. For each of the 7 GEUVADIS replicates (Section 6.2.2), loess curves were fit to estimate per-transcript deviations from overall mean count based on GC content. In each of these plots, each point represents a transcript, with its GC content percent on the x-axis and its read count on the y-axis. As expected, transcripts with high and low GC content tend to be measured as underexpressed.[41,42,105] These models were fit and are illustrated on the $\log_2$ scale: in other words, we added 1 to all transcript counts (to avoid calculating $\log(0)$), log-transformed the counts, then centered the log counts around the mean of all of the log counts. The log-transformation is automatically incorporated in Polyester when adding GC bias.

This level of flexibility is not available with Flux Simulator or BEERS, which only allow specification of the total number of reads per replicate. While it is possible to write custom command-line scripts that induce differential expression using these simulators, differential expression models are built in to Polyester, which is a huge advantage in terms of user-friendliness. This approach offers both a built-in model for convenience and an integrated way to define a custom model for flexibility.

6.2.4 Simulating the RNA Sequencing Process

6.2.4.1 Fragmentation

After the transcripts have been specified and each transcript's abundance in the simulated experiment has been determined by an assigned read count for each replicate, Polyester simulates the RNA sequencing process, described in detail elsewhere,[26] beginning at the fragmentation step. All transcripts present in the experiment are broken into short fragments. There are two options for how fragment lengths can be chosen: the first option is that lengths can be drawn from a normal distribution with mean μ_{fl} and standard deviation σ_{fl}. By default, $\mu_{fl} = 250$ nucleotides and $\sigma_{fl} = 25$, but these parameters can be changed. Alternatively, fragment lengths can be drawn from an empirical length distribution included with the Polyester R package. This empirical distribution (Figure 6.2) was estimated from the insert sizes of the paired-end read alignments of the 7 GEUVADIS replicates described in Section

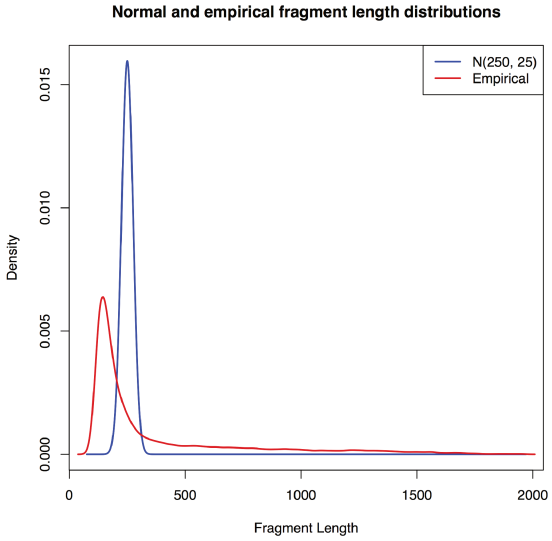

Figure 6.2: Fragment length distributions available in *Polyester*. The red curve shows the fragment length distribution for selected sequencing reads from the GEUVADIS RNA-seq data set; the blue curve shows a normal distribution with mean 250 and standard deviation 25. These two fragment length models are built into the simulator; users can also supply their own.

6.2.2. Insert sizes were calculated using Picard's `CollectInsertSizeMetrics` tool.[106] The empirical density was then estimated using the `logspline` function in R.[107,108] Users can also supply their own fragment length distribution in `logspline` format. This distribution may be estimated from a user's data set or varied to measure the effect of fragment length distribution on downstream results.

Ideally, the fragments generated from a transcript present in the sequencing sample would be uniformly distributed across the transcript. However, coverage across a transcript has been shown to be non-uniform.[13,109,110] In Polyester, users can choose to generate fragments uniformly from transcripts, or they can select one of two possible positional bias models. These models were derived by other researchers,[110] and

they were based on two different fragmentation protocols.

The first model is based on a cDNA fragmentation protocol, and in this model, reads are more likely to come from the 3' end of the transcript being sequenced. The second model incorporates bias caused by a protocol relying on RNA fragmentation, where the middle of each transcript is more likely to be sequenced. Both these models were estimated from Illumina data. Since the exact data from the previous paper[110] was not made available with the manuscript, we extracted the data from Supplementary Figure S3 of the previous paper[110] ourselves, using WebPlotDigitizer,[111] which can estimate the coordinates of data points on a scatterplot given only an image of that scatterplot. For reference, the figure is reproduced here (Figure 6.3), created using the probabilities included as data sets (`cdnaf.rda` and `rnaf.rda`) in the Polyester R package.

6.2.4.2 Sequencing

Polyester simulates unstranded RNA-seq reads in a manner compatible with the Illumina paired-end protocol.[112] In this protocol, read sequences are read off of double-stranded cDNA created from mRNA fragments, separated from other types of RNA using poly-A selection. To mimic this process in Polyester, each fragment selected from an original input transcript is reverse-complemented with probability 0.5: this means the read (for single-end experiments) or mate 1 of the read (for paired-end experiments) is equally likely to have originated from the transcript sequence itself

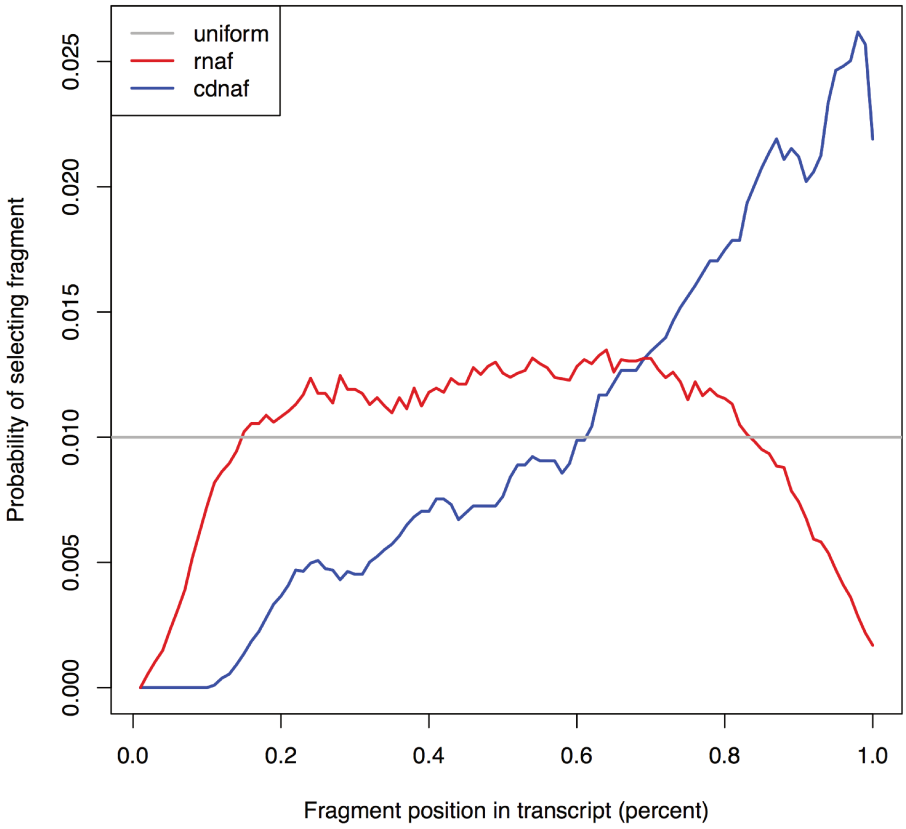

Figure 6.3: Positional bias models implemented in Polyester. The figure aims to replicate a figure previously published as Supplementary Figure S3.[110] Fragment selection across a transcript can be biased based on where the fragment falls in the transcript, as illustrated by the figure. A bias based on RNA fragmentation (`rnaf`) is illustrated in red, while a bias based on cDNA fragmentation (`cdnaf`) is illustrated in blue. The gray line illustrates the uniform, unbiased model, where fragments are equally likely to have originated from any postition in the transcript.

and from the cDNA strand matched to the transcript fragment during sequencing.

Reads are then generated based on these fragments. A single-end read consists of the first R nucleotides of the fragment. For paired-end reads, these first R nucleotides become mate 1, and the last R nucleotides are read off and reverse-complemented to become mate 2. The reverse complementing happens because if mate 1 came from the actual transcript, mate 2 will be read from the complementary cDNA, and if mate 1 came from the complementary cDNA, mate 2 will come from the transcript itself.[113] By default, $R = 100$ and can be adjusted by the user.

Users can choose from a variety of sequencing error models. The simplest one is a uniform error model, where each nucleotide in a read has the same probability p_e of being sequenced incorrectly, and every possible sequencing error is equally likely. For example, if there is an error at a nucleotide which was supposed to be a T, the incorrect base is equally likely to be a G, C, A, or N, where an N is recorded by the sequencing machine if it is unable to call a nucleotide. In the uniform error model, $p_e = 0.005$ by default and can be adjusted.

Several empirical error models are also available in Polyester. These models are based on two dataset-specific models that ship with the `GemSim` software.[114] Separate models are available for a single-end read, mate 1 of a pair, and mate 2 of a pair, from two different sequencing protocols: Illumina Sequencing Kit v4 and TruSeq SBS Kit v5-GA (both from data sequenced on an Illumina Genome Analyzer IIx). These empirical error models include estimated probabilities of making each of the 4 possible

sequencing errors at each position in the read. In general, empirical error probabilities increase toward the end of the read, and mate 2 has higher error probabilities than mate 1 of a pair, and the TruSeq SBS Kit v5-GA error probabilities were lower than the Illumina Sequencing Kit v4 error probabilities (Figures 6.4-6.9).

Polyester can also handle custom error models: users can estimate an error model from their own sequencing data with the `GemErr` utility in `GemSim`. Detailed instructions on how to do this in a way compatible with Polyester are available in the Bioconductor package vignette for Polyester.

After generating sequencing reads and simulating sequencing error, reads are written to disk in FASTA format. The read identifier in the FASTA files specifies the transcript of origin for each read, facilitating assessment of downstream alignment accuracy. Other pertinent simulation information is also automatically written to disk for use in downstream analysis: for each transcript, the transcript name, differential expression status, and fold change is recorded. For each replicate, the file name, group identifier j, and library size factor is recorded.

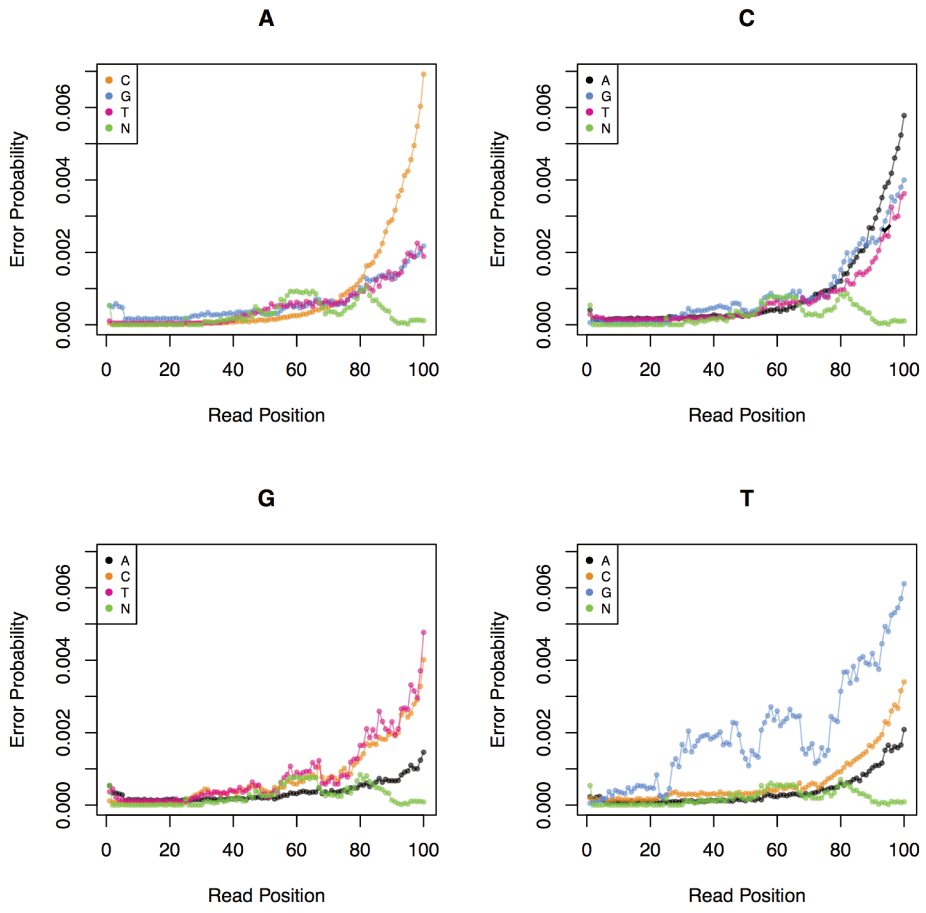

Figure 6.4: Error Model for Illumina Reads (v5), mate 1 of a pair. Empirical error model derived from TruSeq SBS Kit v5-GA chemistry, using Illumina Genome Analyzer IIx, for mate 1 of a paired-end read. Separate panels are shown for each possible true reference nucleotide. Each panel illustrates the probability (y-axis) of mis-sequencing that reference nucleotide in a given read position (x-axis) as any of the 3 other nucleotides, or as an "N" (indicating an "unknown" nucleotide in the read). As expected, error probabilities increase toward the end of the read. If these error models are not suitable, custom error models can be estimated from any set of aligned sequencing reads.

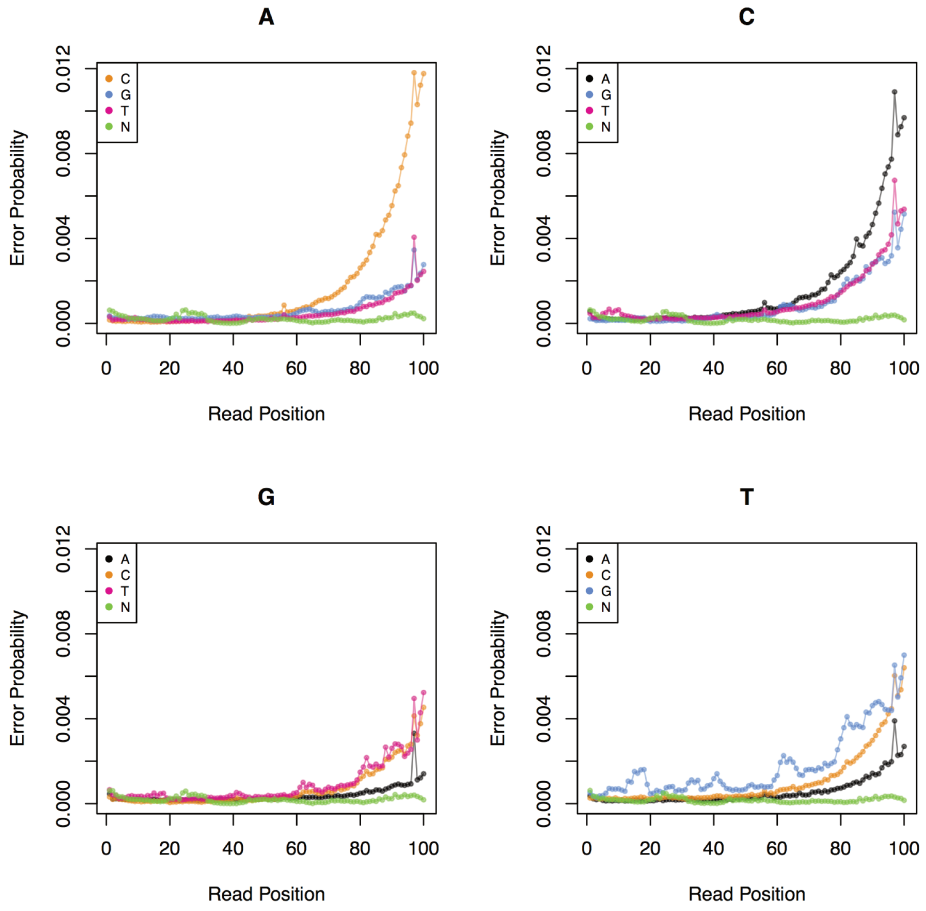

Figure 6.5: Error Model for Illumina Reads (v5), mate 2 of a pair. Empirical error model derived from TruSeq SBS Kit v5-GA chemistry, using Illumina Genome Analyzer IIx, for mate 2 of a paired-end read. Separate panels are shown for each possible true reference nucleotide. Each panel illustrates the probability (y-axis) of mis-sequencing that reference nucleotide in a given read position (x-axis) as any of the 3 other nucleotides, or as an "N" (indicating an "unknown" nucleotide in the read).

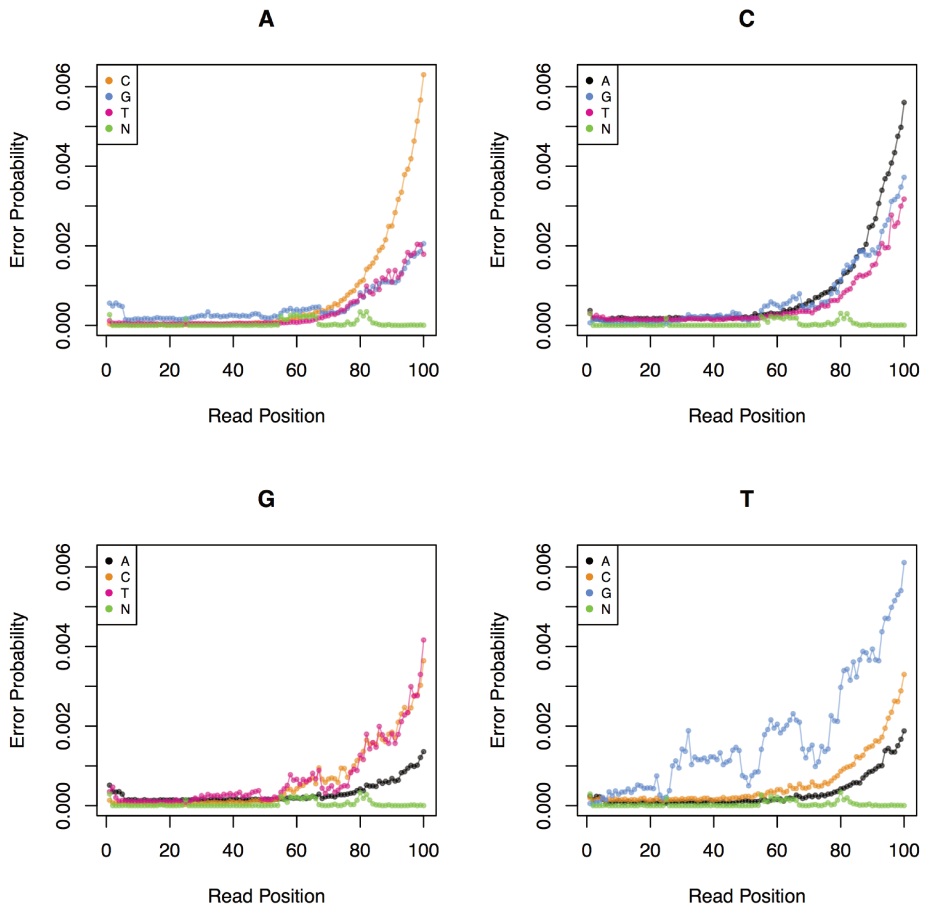

Figure 6.6: Error Model for Illumina Reads (v5), single-end read. Empirical error model derived from TruSeq SBS Kit v5-GA chemistry, using Illumina Genome Analyzer IIx, for a single-end read. Separate panels are shown for each possible true reference nucleotide. Each panel illustrates the probability (y-axis) of mis-sequencing that reference nucleotide in a given read position (x-axis) as any of the 3 other nucleotides, or as an "N" (indicating an "unknown" nucleotide in the read).

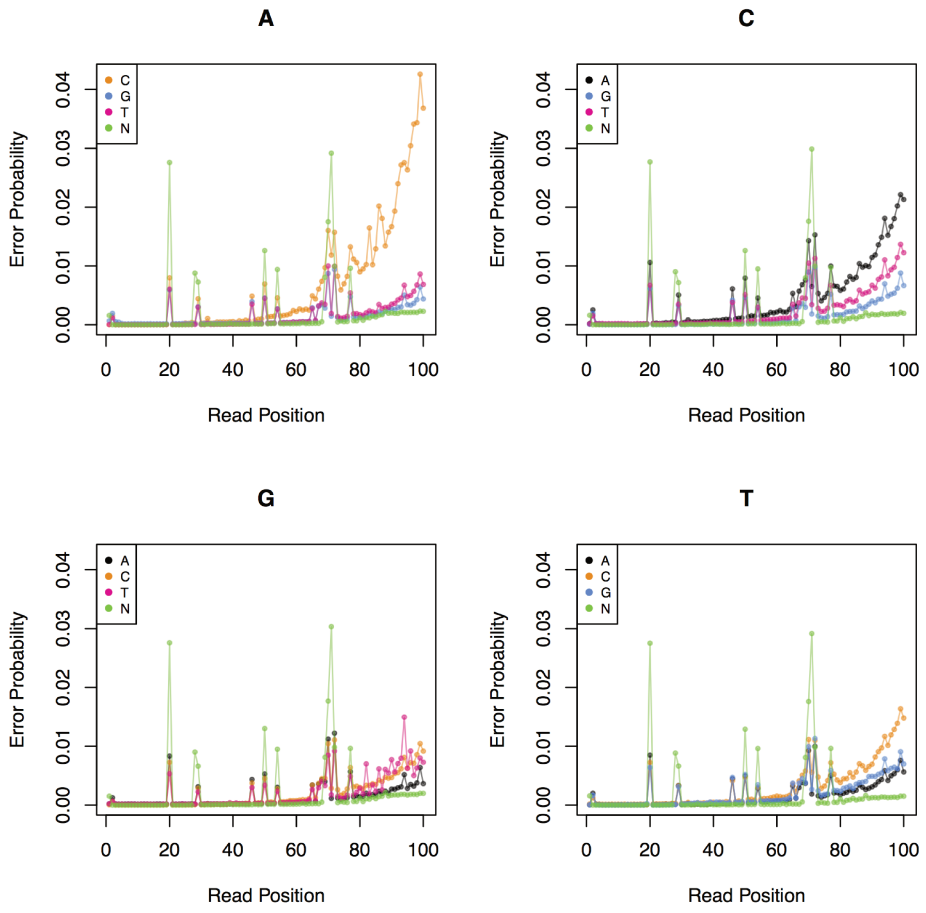

Figure 6.7: Error Model for Illumina Reads (v4), mate 1 of a pair. Empirical error model derived from Illumina Sequencing Kit v4, for mate 1 of a paired-end read. Separate panels are shown for each possible true reference nucleotide. Each panel illustrates the probability (y-axis) of mis-sequencing that reference nucleotide in a given read position (x-axis) as any of the 3 other nucleotides, or as an "N" (indicating an "unknown" nucleotide in the read). This particular sequencing run exhibits some interesting spikes in error rate at different read positions, indicating a possible technical issue with the run; these spikes were also observe in the original analysis of these error rates.[114]

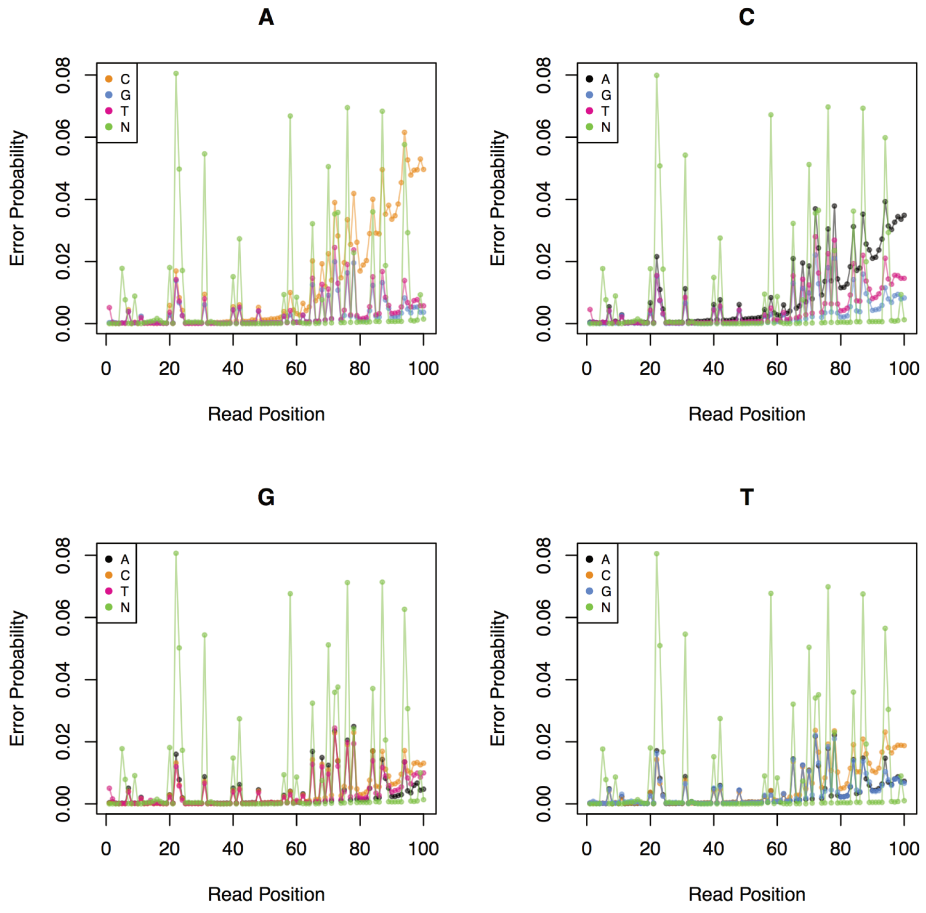

Figure 6.8: Error Model for Illumina Reads (v4), mate 2 of a pair. Empirical error model derived from Illumina Sequencing Kit v4, for mate 2 of a paired-end read. Separate panels are shown for each possible true reference nucleotide. Each panel illustrates the probability (y-axis) of mis-sequencing that reference nucleotide in a given read position (x-axis) as any of the 3 other nucleotides, or as an "N" (indicating an "unknown" nucleotide in the read).

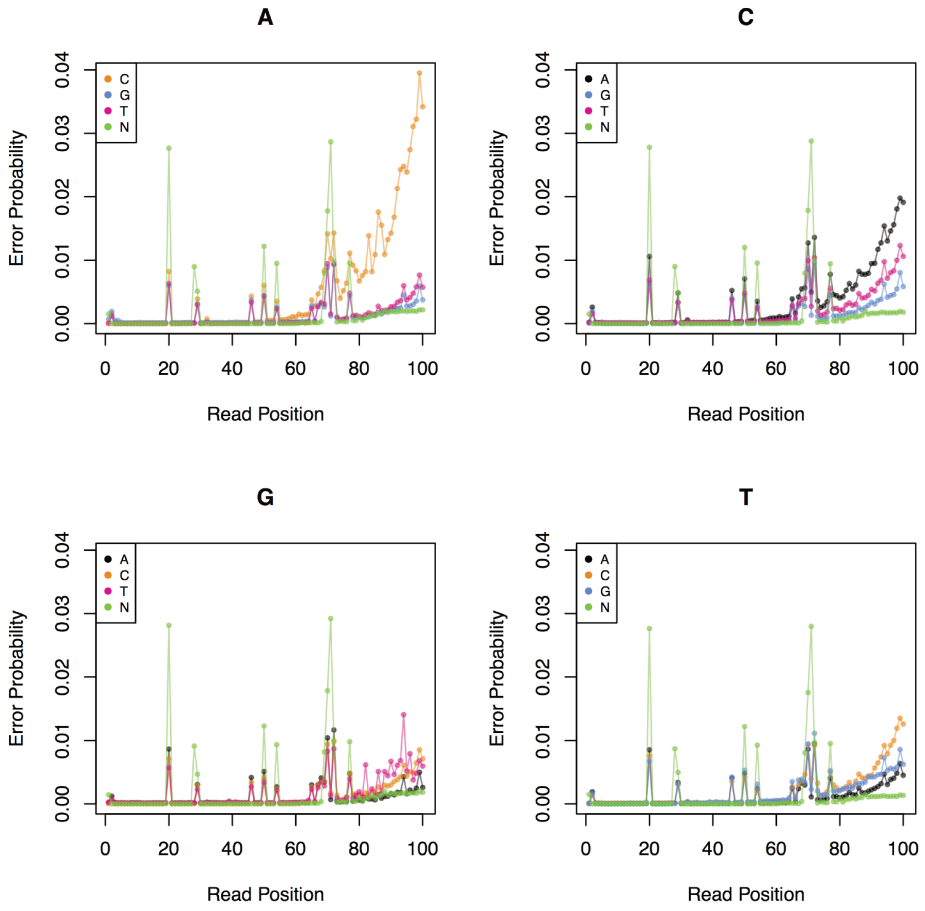

Figure 6.9: Error Model for Illumina Reads (v4), single-end read.Empirical error model derived from Illumina Sequencing Kit v4, a single-end read. Separate panels are shown for each possible true reference nucleotide. Each panel illustrates the probability (y-axis) of mis-sequencing that reference nucleotide in a given read position (x-axis) as any of the 3 other nucleotides, or as an "N" (indicating an "unknown" nucleotide in the read).

6.3 Results

6.3.1 Comparison with Real Data

To show that reads generated with Polyester exhibit realistic properties, we performed a small simulation experiment based on data from the 7 GEUVADIS RNA-seq replicates described in Section 6.2.2. For the experiment, we randomly selected 10 annotated genes with at least one highly-expressed isoform. We relied on the data-driven Cufflinks assembly to determine isoform expression: an annotated gene was considered to have highly-expressed isoforms if at least one of its annotated isoforms overlapped an assembled transcript with an average per-base coverage of at least 20 reads.

The 10 genes that were randomly selected had 15 transcripts between them: two had 3 isoforms, one had 2 isoforms, and the rest had 1 isoform. For the 10 genes, we counted the number of reads overlapping them using the `summarizeOverlaps` function in the Bioconductor package *GenomicAlignments*.[60] Counts were calculated from the TopHat-aligned reads from the GEUVADIS study for the 7 replicates described in Section 6.2.2. We then separated gene counts into isoform-level counts: we calculated per-isoform FPKM values for each of the 15 annotated transcripts using Cufflinks[22] 2.2.1 in its abundance-estimation-only mode, and used the FPKM ratio between isoforms of the same gene to generate isoform-level counts to simulate based on the

gene counts we had already obtained.

We then used these isoform-level counts as input to Polyester, simulating a 7-replicate experiment with the specified number of reads being generated from each of the 15 selected annotated transcripts. Two experiments were simulated: one with all default options (no GC or positional bias, normal fragment length distribution with mean 250 and standard deviation 25, and uniform error model with 0.5% error probability) and one with all default options except for the positional bias model, for which we specified the `rnaf` bias model (Figure 6.3, red line).

The simulated reads were aligned to the hg19 genome with TopHat 2.0.13,[39] and the coverage track for each experiment, for each simulated replicate was compared to the coverage track from the GEUVADIS replicate that generated the simulated replicate's read count. For most of the transcripts, coverage tracks for both experiments looked reasonably similar to the observed coverage track in the GEUVADIS data set (see Figure 6.10 for a representative example).

The simulated coverage tracks were smoother than the coverage track from the GEUVADIS data set, but major trends in the coverage patterns within exons were captured by the simulated reads. There are annotated transcripts for which reads generated by Polyester do not adequately capture the observed coverage in the GEUVADIS data set (Figure 6.11), especially when positional bias is added. This seems to mainly occur in cases where only a very small part of a large exon appears to be expressed in the data set (as is the case in Figure 6.11). The coverage for most of

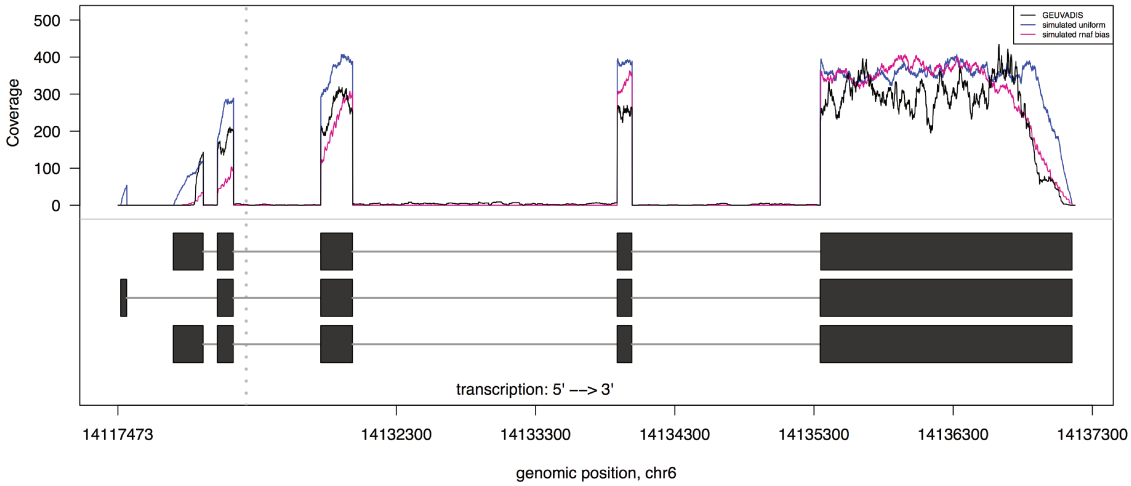

Figure 6.10: Coverage comparison to GEUVADIS data set. We counted the number of reads estimated to have originated from each of these annotated transcripts from gene *CD83* (bottom half of figure) in the GEUVADIS RNA-seq data set, then simulated that same number of reads from each transcript using Polyester and processed those simulated reads. This figure shows the coverage track (y-axis, indicating number of reads with alignments overlapping the specified genomic position) for sample NA06985 (black), reads simulated without positional bias (blue), and read simulated using the `rnaf` bias model (pink). While the simulated coverage tracks look a bit cleaner than the track from the GEUVADIS data set, many of the major within-exon coverage patterns are captured in the simulation, especially with the uniform model. For example, both simulations capture the peak at the beginning of the rightmost exon. *Note*: the gray dotted line indicates that part of a long intron at that location was not illustrated in this plot.

the other transcripts was similar to the real data for most genes and replicates.[115] Reads simulated with `rnaf` bias sometimes had poor coverage for genes consisting of transcripts with many small exons.

For these 15 simulated transcripts, FPKM estimates were positively correlated between each simulated data set and the GEUVADIS data set for each replicate. To get data for this comparison, we used Cufflinks's abundance-estimation-only mode to

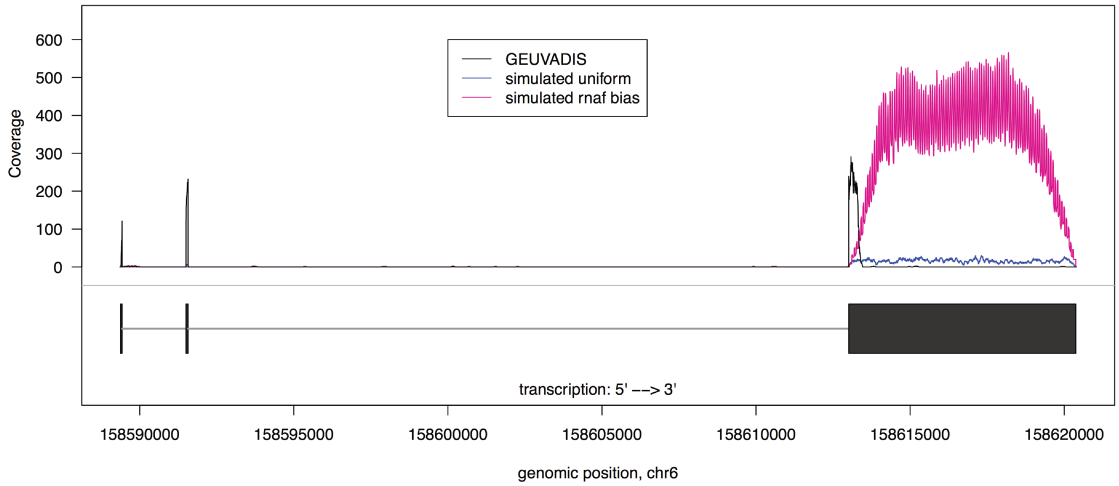

Figure 6.11: Unusual simulated coverage profiles. The bottom panel of this graph illustrates the single isoform of GTF2H5, along with coverage profiles (y-axis) of one replicate from the GEUVADIS data set (black), a data set simulated without positional bias (blue), and a data set simulated with the `rnaf` positional bias model (pink).

get expression estimates for the 15 isoforms based on the simulated reads' alignments, in the same way we calculated expression for the GEUVADIS replicates. We calculated correlation between FPKM estimates of the 15 transcripts for the GEUVADIS data set and for each of the simulated data sets, using correlation instead of absolute FPKM because normalization for number of mapped reads put the sets of FPKMs on different scales.

For the simulation without positional bias, the correlation was extremely high: the minimum correlation across the 7 replicates studied was 0.98. However, the FPKM estimates were less correlated when RNA-fragmentation-related positional bias was induced: all correlations were positive, but weak (Figure 6.12). These results generally

indicate that realistic coverage profiles can be obtained with Polyester but that adding positional bias may cause problems when transcripts have unusual structure. The correlation in FPKM estimates between the simulated data sets and the GEUVADIS samples suggests that Polyester captures transcript-level variation in gene expression data.

6.3.2 Use case: Assessing the accuracy of a differential expression method

To demonstrate a use case for Polyester, we simulated two small differential expression experiments and attempted to discover the simulated differential expression using *limma*.[46]

The first experiment used the default `size` parameter in Polyester, which means the variance of the distribution from which each transcript's count is drawn is equal to 4 times the mean of that distribution. In other words, the mean and variance of the transcript counts are linearly related. We refer to this experiment as "low variance." The second experiment set the `size` parameter to 1 for all transcripts, regardless of the mean count, which means each transcript's mean and variance are quadratically related. This experiment was the "high variance" experiment.

In both scenarios, the main wrapper function in Polyester was used to simulate classic two-group experiments. Reads were simulated from transcripts on human

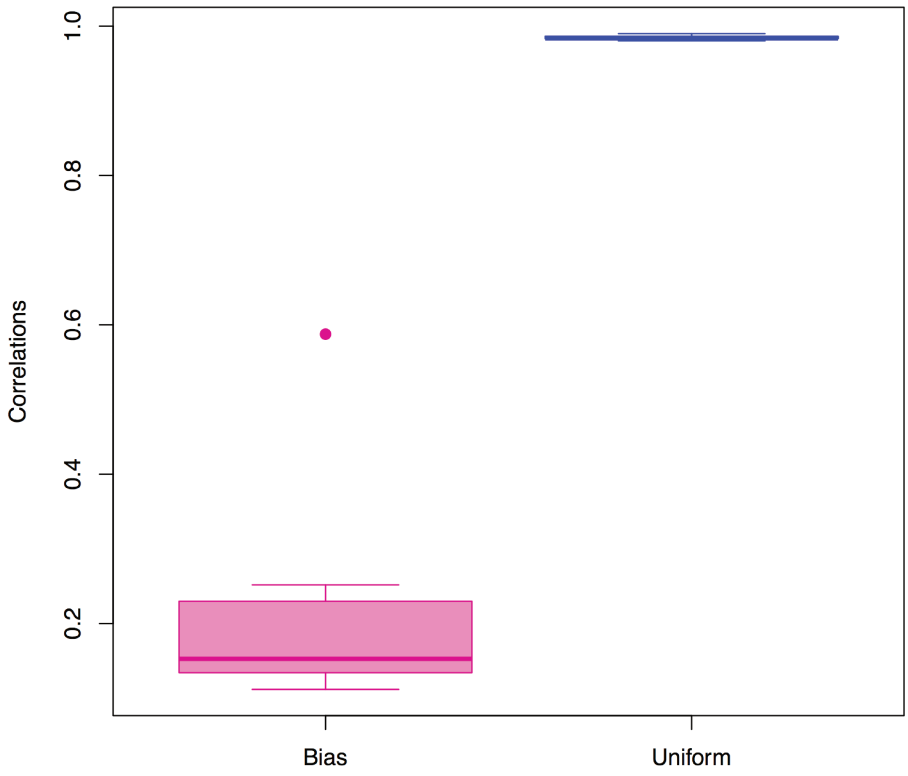

Figure 6.12: Correlations between estimated FPKM values in the GEUVADIS data set compared to simulations, with and without postional bias. Each box in the plot contains 7 correlation measurements, one for each replicate in the study, each of which was obtained by calculating the correlation between FPKM estimates from simulated and GEUVADIS data for the 15 transcripts in the study. Correlations for the simulation with positional bias are shown on the left, and for the simulation with uniform fragmentation (no positional bias) on the right. Correlations were all positive, but were weak in the simulation with bias and very strong in the simulation without bias.

chromosome 22 (hg19 build, $N = 926$). μ_k was set to length(transcript$_k$)/5, which corresponds to approximately 20x coverage for reads of length 100. We randomly chose 75 transcripts to have $\lambda = 3$ and 75 to have $\lambda = 1/3$; the rest had $\lambda = 1$. For $n_j = 7$ replicates in each group j, we simulated paired-end reads from 250-base fragments ($\sigma_{fl} = 25$), with a uniform error probability and the default error rate of 0.005. Simulated reads were aligned to hg19 with TopHat 2.0.13,[39] and Cufflinks 2.2.1[22] was used to obtain expression estimates for the 926 transcripts from which transcripts were simulated. Expression was measured using FPKM (fragments per kilobase per million mapped reads). We then ran transcript-level differential expression tests using *limma*.[46] Specifically, for each transcript k, the following linear model was fit:

$$\log_2(\text{FPKM}_k + 1) = \alpha_k + \beta_k X_j + \gamma_k W_j$$

where FPKM_k is the expression measurement for transcript k, X_j is 0 or 1 depending on which group sample j was assigned to, and W_j is a library-size adjustment, defined as the 75th percentile over all k of the $\log_2(\text{FPKM}_k + 1)$ values for replicate j.[86] We fit these linear models for each transcript, and for each β_k, we calculated moderated t-statistics and associated p-values using the shrinkage methodology in *limma*'s `eBayes` function. We calculated ROC curves based on these p-values and our knowledge of the true differential expression status of each transcript. Sensitivity and specificity of the *limma* differential expression analysis were high for the small-variance scenario, but were diminished in the large-variance scenario, as expected

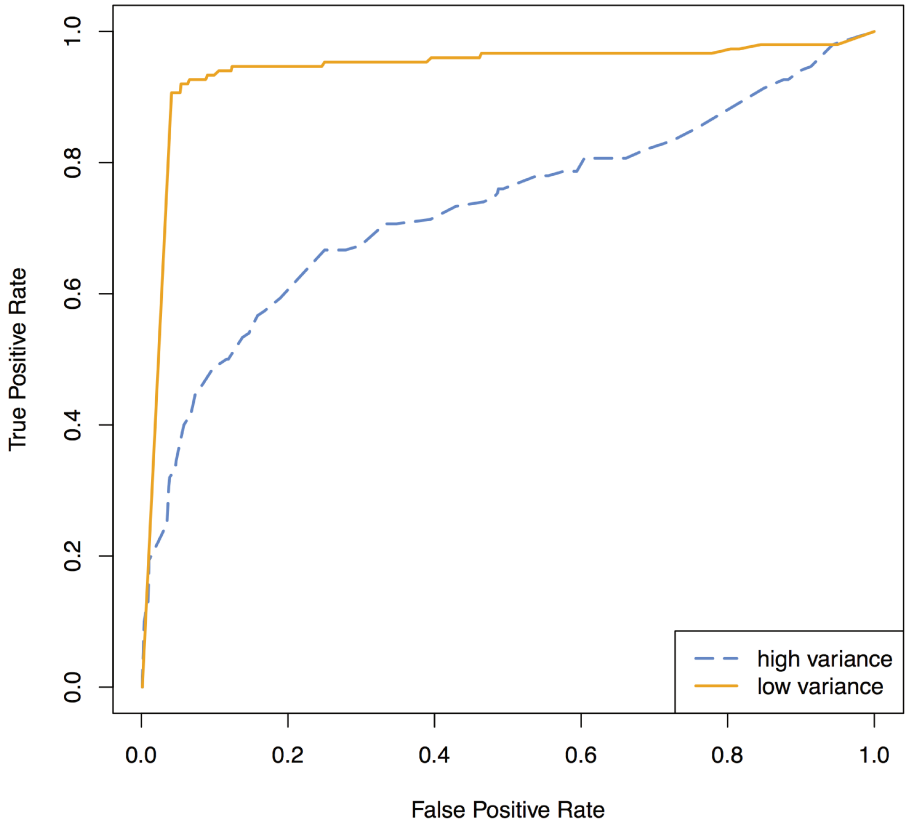

Figure 6.13: ROC curves for transcript-level differential expression calls from Polyester data sets. For varying significance (p- or q-value) cutoffs, sensitivity and specificity from the simulation experiments. Differential expression was more difficult to detect under conditions where expression levels were highly variable between replicates, as expected.

(Figure 6.13).

Since expression fold changes can be explicitly specified in Polyester, we can also investigate whether those fold changes are preserved throughout this RNA-seq data analysis pipeline (Figure 6.14). In general, the coefficient distributions for transcripts not specified to be differentially expressed were centered around zero, as expected,

since models were fit on the log scale. The coefficient distributions should have been centered around $\log_2(3) = 1.58$ for the overexpressed transcripts (expression level three times higher in the first group), and around $\log_2(1/3) = -1.58$ for the underexpressed transcripts (expression level three times higher in the second group). The overexpressed distributions had means 1.39 and 1.44 in the high- and low-variance scenarios, respectively, and the underexpressed distributions had means -1.57 and -1.60 in the high- and low-variance scenarios, respectively. Coefficient estimates were much more variable in the scenario with higher expression variance (Figure 6.14). These numbers are similar to the the specified value of 1.58, indicating that the RNA-seq pipeline used to analyze these data sets satisfactorily captured the existence and magnitude of the differential expression set in the experiment simulated with Polyester.

For this differential experiment, where about 639,000 reads per sample were simulated, read generation took 1-2 minutes per biological replicate in the experiment and 4.4G memory was used on a single cluster node with one core.

These examples illustrate some of the many possible ways Polyester can be used to explore the effects of analysis choices on downstream differential expression results.

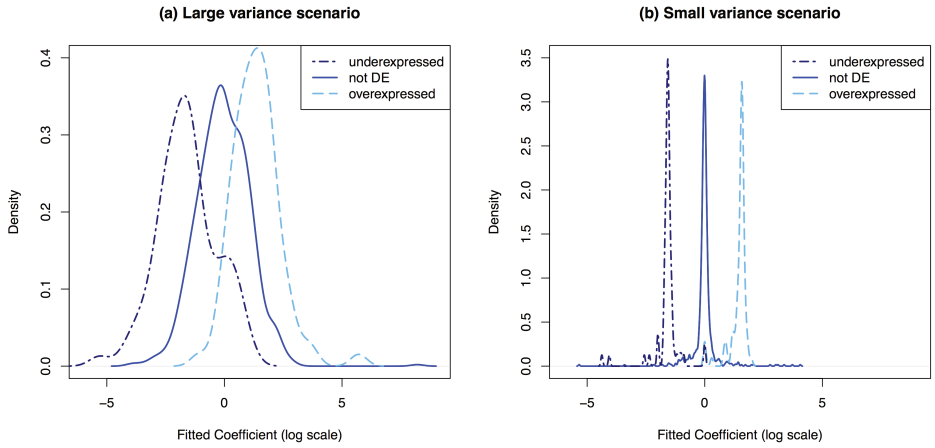

Figure 6.14: Coefficient distributions from differential expression models. Distributions from the high-variance scenario are shown in panel (a) and from the low-variance scenario are shown in panel (b). These distributions of estimated log fold changes between the two simulation groups tend to be centered around the values specified at the beginning of the simulation, and there is more variability in the coefficient estimates for high-variance scenario, as expected.

6.4 Discussion

In this paper, we propose a lightweight, flexible RNA-seq read simulator allowing users to set differential expression levels at the isoform level. A full experiment with biological replicates can be simulated with one command, and time-consuming alignment is not required beforehand.

The sequencing process is complex, and some subtleties and potential biases present in that process are not yet implemented in Polyester but could be in the future. For example, adding random hexamer priming bias,[116] implementing PCR amplification bias[117] or other biases that depend on the specific nucleotides being sequenced, simulating quality scores for base calls, and adding the ability to simulate

indels are all possibilities for future improvements. However, our comparisons with real data suggest that the Polyester model sufficiently mimicks real sequencing data to be practically useful.

6.5 Software

Polyester is available from Bioconductor: `http://bioconductor.org/packages/release/bioc/html/polyester.html`. The development version is available on GitHub: `https://github.com/alyssafrazee/polyester`. Community contributions and bug reports are welcomed in the development version. Code for the analysis shown in this chapter is available at `https://github.com/alyssafrazee/polyester_code`.

6.6 Acknowledgements

For this project, JL and BL were supported by NIH R01 GM105705. AF was supported by a Hopkins Sommer Scholarship. AJ was supported by the Lieber Institute for Brain Development.

Bibliography

[1] F. H. Crick, "On protein synthesis." in *Symposia of the Society for Experimental Biology*, vol. 12, 1958, p. 138.

[2] ——, "Central dogma of molecular biology," *Nature*, vol. 227, no. 5258, pp. 561–563, 1970.

[3] B. E. Stranger, A. C. Nica, M. S. Forrest, A. Dimas, C. P. Bird, C. Beazley, C. E. Ingle, M. Dunning, P. Flicek, D. Koller *et al.*, "Population genomics of human gene expression," *Nature Genetics*, vol. 39, no. 10, pp. 1217–1224, 2007.

[4] D. T. Ross, U. Scherf, M. B. Eisen, C. M. Perou, C. Rees, P. Spellman, V. Iyer, S. S. Jeffrey, M. Van de Rijn, M. Waltham *et al.*, "Systematic variation in gene expression patterns in human cancer cell lines," *Nature Genetics*, vol. 24, no. 3, pp. 227–235, 2000.

[5] A. Ralston and K. Shaw, "Gene expression regulates cell differentiation," *Nature Education*, vol. 1, p. 127, 2008.

BIBLIOGRAPHY

[6] J. C. Alwine, D. J. Kemp, and G. R. Stark, "Method for detection of specific RNAs in agarose gels by transfer to diazobenzyloxymethyl-paper and hybridization with DNA probes," *Proceedings of the National Academy of Sciences*, vol. 74, no. 12, pp. 5350–5354, 1977.

[7] M. D. Adams, J. M. Kelley, J. D. Gocayne, M. Dubnick, M. H. Polymeropoulos, H. Xiao, C. R. Merril, A. Wu, B. Olde, R. F. Moreno *et al.*, "Complementary DNA sequencing: expressed sequence tags and human genome project," *Science*, vol. 252, no. 5013, pp. 1651–1656, 1991.

[8] M. A. Abbott, B. J. Poiesz, B. C. Byrne, S. Kwok, J. J. Sninsky, and G. D. Ehrlich, "Enzymatic gene amplification: qualitative and quantitative methods for detecting proviral DNA amplified in vitro," *Journal of Infectious Diseases*, vol. 158, no. 6, pp. 1158–1169, 1988.

[9] A.-C. Syvänen, M. Bengtström, T. Jukka, and H. Söderlund, "Quantification of polymerase chain reaction products by affinity-based hybrid collection," *Nucleic Acids Research*, vol. 16, no. 23, pp. 11 327–11 338, 1988.

[10] L. T. Chow, R. E. Gelinas, T. R. Broker, and R. J. Roberts, "An amazing sequence arrangement at the 5 ends of adenovirus 2 messenger RNA," *Cell*, vol. 12, no. 1, pp. 1–8, 1977.

[11] S. M. Berget, C. Moore, and P. A. Sharp, "Spliced segments at the 5'terminus

of adenovirus 2 late mRNA," *Proceedings of the National Academy of Sciences*, vol. 74, no. 8, pp. 3171–3175, 1977.

[12] D. L. Black, "Mechanisms of alternative pre-messenger rna splicing," *Annual Review of Biochemistry*, vol. 72, no. 1, pp. 291–336, 2003.

[13] A. Mortazavi, B. A. Williams, K. McCue, L. Schaeffer, and B. Wold, "Mapping and quantifying mammalian transcriptomes by RNA-seq," *Nature Methods*, vol. 5, no. 7, pp. 621–628, 2008.

[14] Y. Katz, E. Wang, E. Airoldi, and C. Burge, "Analysis and design of RNA sequencing experiments for identifying isoform regulation," *Nature Methods*, vol. 7, no. 12, pp. 1009–1015, 2010.

[15] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown, "Quantitative monitoring of gene expression patterns with a complementary DNA microarray," *Science*, vol. 270, no. 5235, pp. 467–470, 1995.

[16] G. Smyth *et al.*, "Linear models and empirical Bayes methods for assessing differential expression in microarray experiments." *Statistical Applications in Genetics and Molecular Biology*, vol. 3, no. 1, p. 3, 2004.

[17] R. A. Irizarry, B. Hobbs, F. Collin, Y. D. Beazer-Barclay, K. J. Antonellis, U. Scherf, and T. P. Speed, "Exploration, normalization, and summaries of

high density oligonucleotide array probe level data," *Biostatistics*, vol. 4, no. 2, pp. 249–264, 2003.

[18] S. Dudoit, Y. Yang, M. Callow, and T. Speed, "Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments," *Statistica Sinica*, vol. 12, no. 1, pp. 111–140, 2002.

[19] U. Nagalakshmi, Z. Wang, K. Waern, C. Shou, D. Raha, M. Gerstein, and M. Snyder, "The transcriptional landscape of the yeast genome defined by RNA sequencing," *Science*, vol. 320, no. 5881, pp. 1344–1349, 2008.

[20] M. Guttman, M. Garber, J. Levin, J. Donaghey, J. Robinson, X. Adiconis, L. Fan, M. Koziol, A. Gnirke, C. Nusbaum, J. Rinn, E. Lander, and A. Regev, "Ab initio reconstruction of transcriptomes of pluripotent and lineage committed cells reveals gene structures of thousands of lincRNAs," *Nature Biotechnology*, vol. 28, no. 5, pp. 503–510, May 2010.

[21] M. Clark, P. Amaral, F. Schlesinger, M. Dinger, R. Taft, J. Rinn, C. Ponting, P. Stadler, K. Morris, A. Morillon *et al.*, "The reality of pervasive transcription," *PLoS Biology*, vol. 9, no. 7, p. e1000625, 2011.

[22] C. Trapnell, B. A. Williams, G. Pertea, A. Mortazavi, G. Kwan, M. J. van Baren, S. L. Salzberg, B. J. Wold, and L. Pachter, "Transcript assembly and quantification by RNA-seq reveals unannotated transcripts and isoform switch-

ing during cell differentiation," *Nature Biotechnology*, vol. 28, no. 5, pp. 511–515, 2010.

[23] L. Stein *et al.*, "The case for cloud computing in genome informatics," *Genome Biology*, vol. 11, no. 5, p. 207, 2010.

[24] C. Trapnell, D. G. Hendrickson, M. Sauvageau, L. Goff, J. L. Rinn, and L. Pachter, "Differential analysis of gene regulation at transcript resolution with RNA-seq," *Nature Biotechnology*, vol. 31, no. 1, pp. 46–53, 2013.

[25] R. C. Gentleman, V. J. Carey, D. M. Bates, and others, "Bioconductor: Open software development for computational biology and bioinformatics," *Genome Biology*, vol. 5, p. R80, 2004. [Online]. Available: http://genomebiology.com/2004/5/10/R80

[26] A. Oshlack, M. D. Robinson, M. D. Young *et al.*, "From RNA-seq reads to differential expression results," *Genome Biology*, vol. 11, no. 12, p. 220, 2010.

[27] M. D. Robinson, D. J. McCarthy, and G. K. Smyth, "edgeR: a Bioconductor package for differential expression analysis of digital gene expression data," *Bioinformatics*, vol. 26, no. 1, pp. 139–140, 2010.

[28] D. J. McCarthy, Y. Chen, and G. K. Smyth, "Differential expression analysis of multifactor RNA-seq experiments with respect to biological variation," *Nucleic Acids Research*, vol. 40, no. 10, pp. 4288–4297, 2012.

BIBLIOGRAPHY

[29] S. Anders and W. Huber, "Differential expression analysis for sequence count data," *Genome Biology*, vol. 11, no. 10, p. R106, 2010.

[30] M. Griffith, O. Griffith, J. Mwenifumbo, R. Goya, A. Morrissy, R. Morin, R. Corbett, M. Tang, Y. Hou, T. Pugh *et al.*, "Alternative expression analysis by RNA sequencing," *Nature Methods*, vol. 7, no. 10, pp. 843–847, 2010.

[31] S. Anders, A. Reyes, and W. Huber, "Detecting differential usage of exons from RNA-seq data," *Genome Research*, vol. 22, no. 10, pp. 2008–2017, 2012.

[32] W. Wang, Z. Qin, Z. Feng, X. Wang, and X. Zhang, "Identifying differentially spliced genes from two groups of RNA-seq samples," *Gene*, vol. 518, pp. 164–170, 2012.

[33] W. Klimke, C. O'Donovan, O. White, J. Brister, K. Clark, B. Fedorov, I. Mizrachi, K. Pruitt, and T. Tatusova, "Solving the problem: Genome annotation standards before the data deluge," *Standards in Genomic Sciences*, vol. 5, no. 1, p. 168, 2011.

[34] J. H. Bullard, E. Purdom, K. D. Hansen, and S. Dudoit, "Evaluation of statistical methods for normalization and differential expression in mRNA-seq experiments," *BMC Bioinformatics*, vol. 11, no. 1, p. 94, 2010.

[35] W. Li, J. Feng, and T. Jiang, "Isolasso: a lasso regression approach to RNA-

seq based transcriptome assembly," *Journal of Computational Biology*, vol. 18, no. 11, pp. 1693–1707, 2011.

[36] R. Patro, S. M. Mount, and C. Kingsford, "Sailfish enables alignment-free isoform quantification from RNA-seq reads using lightweight algorithms," *Nature Biotechnology*, vol. 32, no. 5, pp. 462–464, 2014.

[37] A. Leśniewska and M. Okoniewski, "rnaSeqMap: a Bioconductor package for RNA sequencing data exploration," *BMC Bioinformatics*, vol. 12, no. 1, p. 200, 2011.

[38] O. Stegle, P. Drewe, R. Bohnert, K. Borgwardt, and G. Rätsch, "Statistical tests for detecting differential RNA-transcript expression from read counts," *Nature Precedings*, 2010.

[39] C. Trapnell, L. Pachter, and S. L. Salzberg, "TopHat: discovering splice junctions with RNA-seq," *Bioinformatics*, vol. 25, no. 9, pp. 1105–1111, 2009.

[40] G. E. Box and D. R. Cox, "An analysis of transformations," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 211–252, 1964.

[41] K. D. Hansen, R. A. Irizarry, and W. Zhijin, "Removing technical variability in RNA-seq data using conditional quantile normalization," *Biostatistics*, vol. 13, no. 2, pp. 204–216, 2012.

BIBLIOGRAPHY

[42] D. Risso, K. Schwartz, G. Sherlock, and S. Dudoit, "GC-content normalization for RNA-seq data," *BMC Bioinformatics*, vol. 12, no. 1, p. 480, 2011.

[43] B. Efron, "Microarrays, empirical Bayes and the two-groups model," *Statistical Science*, vol. 23, no. 1, pp. 1–22, 2008.

[44] A. Jaffe, P. Murakami, H. Lee, J. Leek, M. Fallin, A. Feinberg, and R. Irizarry, "Bump hunting to identify differentially methylated regions in epigenetic epidemiology studies," *International Journal of Epidemiology*, vol. 41, no. 1, pp. 200–209, 2012.

[45] K. Hansen, Z. Wu, R. Irizarry, and J. Leek, "Sequencing technology does not eliminate biological variability," *Nature Biotechnology*, vol. 29, no. 7, pp. 572–573, 2011.

[46] G. K. Smyth, "Limma: linear models for microarray data," in *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*, R. Gentleman, V. Carey, S. Dudoit, R. Irizarry, and W. Huber, Eds. New York: Springer, 2005, pp. 397–420.

[47] C. Fraley and A. Raftery, "Model-based clustering, discriminant analysis, and density estimation," *Journal of the American Statistical Association*, vol. 97, no. 458, pp. 611–631, 2002.

BIBLIOGRAPHY

[48] G. Forney Jr, "The Viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[49] D. Harte, *HiddenMarkov: Hidden Markov Models*, Statistics Research Associates, Wellington, 2012, R package version 1.7-0. [Online]. Available: http://cran.at.r-project.org/web/packages/HiddenMarkov

[50] L. Collado-Torres, A. Frazee, A. Jaffe, and J. Leek, *derfinder: Annotation-agnostic differential expression analysis of RNA-seq data at base-pair resolution*, 2014, R package version 1.0.10. [Online]. Available: http://www.bioconductor.org/packages/release/bioc/html/derfinder.html

[51] A. B. Olshen, E. Venkatraman, R. Lucito, and M. Wigler, "Circular binary segmentation for the analysis of array-based DNA copy number data," *Biostatistics*, vol. 5, no. 4, pp. 557–572, 2004.

[52] J. D. Storey and R. Tibshirani, "Statistical significance for genomewide studies," *Proceedings of the National Academy of Sciences*, vol. 100, no. 16, pp. 9440–9445, 2003.

[53] S. Dudoit and M. J. Van Der Laan, *Multiple testing procedures with applications to genomics*. Springer, 2008.

[54] P. Westfall, S. Young, and S. P. Wright, "On adjusting p-values for multiplicity," *Biometrics*, vol. 49, no. 3, pp. 941–945, 1993.

BIBLIOGRAPHY

[55] J. T. Leek and J. D. Storey, "The joint null criterion for multiple hypothesis tests," *Statistical Applications in Genetics and Molecular Biology*, vol. 10, no. 1, 2011.

[56] C. R. Genovese, N. A. Lazar, and T. Nichols, "Thresholding of statistical maps in functional neuroimaging using the false discovery rate," *Neuroimage*, vol. 15, no. 4, pp. 870–878, 2002.

[57] T. E. Nichols and A. P. Holmes, "Nonparametric permutation tests for functional neuroimaging: a primer with examples," *Human Brain Mapping*, vol. 15, no. 1, pp. 1–25, 2002.

[58] Illumina, "Illumina iGenomes," http://ccb.jhu.edu/software/tophat/igenomes.shtml, 2012.

[59] M. Morgan and H. Pagès, *Rsamtools: Binary alignment (BAM), variant call (BCF), or tabix file import*, R package version 1.6.3. [Online]. Available: http://bioconductor.org/packages/release/bioc/html/Rsamtools.html

[60] M. Lawrence, W. Huber, H. Pagès, P. Aboyoun, M. Carlson, R. Gentleman, M. Morgan, and V. Carey, "Software for computing and annotating genomic ranges," *PLoS Computational Biology*, vol. 9, 2013. [Online]. Available: http://www.ploscompbiol.org/article/info%3Adoi%2F10.1371%2Fjournal.pcbi.1003118

BIBLIOGRAPHY

[61] S. Durinck, Y. Moreau, A. Kasprzyk, S. Davis, B. De Moor, A. Brazma, and W. Huber, "BioMart and Bioconductor: a powerful link between biological databases and microarray data analysis," *Bioinformatics*, vol. 21, no. 16, pp. 3439–3440, 2005.

[62] T. Griebel, B. Zacher, P. Ribeca, E. Raineri, V. Lacroix, R. Guigó, and M. Sammeth, "Modelling and simulating generic RNA-seq experiments with the flux simulator," *Nucleic Acids Research*, vol. 40, no. 20, pp. 10 073–10 083, 2012.

[63] C. Trapnell, A. Roberts, L. Goff, G. Pertea, D. Kim, D. R. Kelley, H. Pimentel, S. L. Salzberg, J. L. Rinn, and L. Pachter, "Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks," *Nature Protocols*, vol. 7, no. 3, pp. 562–578, 2012.

[64] M. G. Grabherr, B. J. Haas, M. Yassour, J. Z. Levin, D. A. Thompson, I. Amit, X. Adiconis, L. Fan, R. Raychowdhury, Q. Zeng *et al.*, "Full-length transcriptome assembly from RNA-seq data without a reference genome," *Nature Biotechnology*, vol. 29, no. 7, pp. 644–652, 2011.

[65] M. Dermitzakis, G. Getz, K. Ardle, R. Guigo, and for the GTEx consortium, "Response to: "GTEx is throwing away 90% of their data"," http://liorpachter.wordpress.com/2013/10/31/response-to-gtex-is-throwing-away-90-of-their-data/, 2013.

[66] M. Pertea, G. M. Pertea, C. M. Antonescu, T.-C. Chang, J. Mendell, and S. L.

Salzberg, "Stringtie enables reconstruction of a transcriptome from RNA-seq reads," *Nature Biotechnology*, 2015.

[67] B. Li and C. N. Dewey, "RSEM: accurate transcript quantification from RNA-seq data with or without a reference genome," *BMC Bioinformatics*, vol. 12, no. 1, p. 323, 2011.

[68] C. W. Law, Y. Chen, W. Shi, and G. K. Smyth, "Voom: precision weights unlock linear model analysis tools for RNA-seq read counts," *Genome Biology*, vol. 15, no. 2, p. R29, 2014.

[69] Y. Chen, D. McCarthy, M. Robinson, and G. K. Smyth, "edgeR: differential expression analysis of digital gene expression data users guide," http://www.bioconductor.org/packages/release/bioc/vignettes/edgeR/inst/doc/edgeRUsersGuide.pdf, 2014.

[70] N. Leng, J. A. Dawson, J. A. Thomson, V. Ruotti, A. I. Rissman, B. M. Smits, J. D. Haag, M. N. Gould, R. M. Stewart, and C. Kendziorski, "EBSeq: an empirical Bayes hierarchical model for inference in RNA-seq experiments," *Bioinformatics*, vol. 29, no. 8, pp. 1035–1043, 2013.

[71] B. Langmead, C. Trapnell, M. Pop, and S. L. Salzberg, "Ultrafast and memory-efficient alignment of short DNA sequences to the human genome," *Genome Biology*, vol. 10, no. 3, p. R25, 2009.

[72] R. Lister, M. Pelizzola, Y. S. Kida, R. D. Hawkins, J. R. Nery, G. Hon, J. Antosiewicz-Bourget, R. O'Malley, R. Castanon, S. Klugman *et al.*, "Hotspots of aberrant epigenomic reprogramming in human induced pluripotent stem cells," *Nature*, vol. 471, no. 7336, pp. 68–73, 2011.

[73] R. Lister, E. A. Mukamel, J. R. Nery, M. Urich, C. A. Puddifoot, N. D. Johnson, J. Lucero, Y. Huang, A. J. Dwork, M. D. Schultz *et al.*, "Global epigenomic reconfiguration during mammalian brain development," *Science*, vol. 341, no. 6146, 2013.

[74] R. S. Young, A. C. Marques, C. Tibbit, W. Haerty, A. R. Bassett, J.-L. Liu, and C. P. Ponting, "Identification and properties of 1,119 candidate lincrna loci in the drosophila melanogaster genome," *Genome Biology and Evolution*, vol. 4, no. 4, pp. 427–442, 2012.

[75] S. Djebali, C. A. Davis, A. Merkel, A. Dobin, T. Lassmann, A. Mortazavi, A. Tanzer, J. Lagarde, W. Lin, F. Schlesinger *et al.*, "Landscape of transcription in human cells," *Nature*, vol. 489, no. 7414, pp. 101–108, 2012.

[76] B. R. Graveley, A. N. Brooks, J. W. Carlson, M. O. Duff, J. M. Landolin, L. Yang, C. G. Artieri, M. J. van Baren, N. Boley, B. W. Booth *et al.*, "The developmental transcriptome of drosophila melanogaster," *Nature*, vol. 471, no. 7339, pp. 473–479, 2011.

[77] P. AC't Hoen, M. R. Friedländer, J. Almlöf, M. Sammeth, I. Pulyakhina, S. Y.

Anvar, J. F. Laros, H. P. Buermans, O. Karlberg, M. Brännvall *et al.*, "Reproducibility of high-throughput mRNA and small RNA sequencing across laboratories," *Nature Biotechnology*, 2013.

[78] T. Lappalainen, M. Sammeth, M. R. Friedländer, P. ACt Hoen, J. Monlong, M. A. Rivas, M. Gonzàlez-Porta, N. Kurbatova, T. Griebel, P. G. Ferreira *et al.*, "Transcriptome and genome sequencing uncovers functional variation in humans," *Nature*, 2013.

[79] A. C. Frazee, S. Sabunciyan, K. D. Hansen, R. A. Irizarry, and J. T. Leek, "Differential expression analysis of RNA-seq data at single-base resolution," *Biostatistics*, vol. 15, pp. 413–426, 2014.

[80] S. C. Kim, Y. Jung, J. Park, S. Cho, C. Seo, J. Kim, P. Kim, J. Park, J. Seo, J. Kim *et al.*, "A high-dimensional, deep-sequencing study of lung adenocarcinoma in female never-smokers," *PloS one*, vol. 8, no. 2, p. e55596, 2013.

[81] L. Yan, M. Yang, H. Guo, L. Yang, J. Wu, R. Li, P. Liu, Y. Lian, X. Zheng, J. Yan *et al.*, "Single-cell RNA-seq profiling of human preimplantation embryos and embryonic stem cells," *Nature Structural & Molecular Biology*, 2013.

[82] A. Coletta, C. Molter, R. Duqué, D. Steenhoff, J. Taminau, V. De Schaetzen, S. Meganck, C. Lazar, D. Venet, V. Detours *et al.*, "InSilico DB genomic datasets hub: an efficient starting point for analyzing genome-wide studies in

genepattern, integrative genomics viewer, and r/bioconductor," *Genome Biology*, vol. 13, no. 11, p. R104, 2012.

[83] R. Leinonen, H. Sugawara, and M. Shumway, "The Sequence Read Archive," *Nucleic Acids Research*, p. gkq1019, 2010.

[84] A. Schroeder, O. Mueller, S. Stocker, R. Salowsky, M. Leiber, M. Gassmann, S. Lightfoot, W. Menzel, M. Granzow, and T. Ragg, "The RIN: an RNA integrity number for assigning integrity values to RNA measurements," *BMC Molecular Biology*, vol. 7, no. 1, p. 3, 2006.

[85] J. D. Storey, W. Xiao, J. T. Leek, R. G. Tompkins, and R. W. Davis, "Significance analysis of time course microarray experiments," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 36, pp. 12 837–12 842, 2005.

[86] J. N. Paulson, O. C. Stine, H. C. Bravo, and M. Pop, "Differential abundance analysis for microbial marker-gene surveys," *Nature Methods*, 2013.

[87] A. A. Shabalin, "Matrix eQTL: ultra fast eQTL analysis via large matrix operations," *Bioinformatics*, vol. 28, no. 10, pp. 1353–1358, 2012.

[88] A. L. Price, N. J. Patterson, R. M. Plenge, M. E. Weinblatt, N. A. Shadick, and D. Reich, "Principal components analysis corrects for stratification in genome-wide association studies," *Nature Genetics*, vol. 38, no. 8, pp. 904–909, 2006.

[89] S. Purcell, B. Neale, K. Todd-Brown, L. Thomas, M. A. Ferreira, D. Bender, J. Maller, P. Sklar, P. I. De Bakker, M. J. Daly *et al.*, "PLINK: a tool set for whole-genome association and population-based linkage analyses," *The American Journal of Human Genetics*, vol. 81, no. 3, pp. 559–575, 2007.

[90] J. T. Leek and J. D. Storey, "Capturing heterogeneity in gene expression studies by surrogate variable analysis," *PLoS Genetics*, vol. 3, no. 9, p. e161, 2007.

[91] B. Devlin and K. Roeder, "Genomic control for association studies," *Biometrics*, vol. 55, no. 4, pp. 997–1004, 1999.

[92] B. Li, V. Ruotti, R. M. Stewart, J. A. Thomson, and C. N. Dewey, "RNA-seq gene expression estimation with read mapping uncertainty," *Bioinformatics*, vol. 26, no. 4, pp. 493–500, 2010.

[93] G. P. Wagner, K. Kin, and V. J. Lynch, "Measurement of mRNA abundance using RNA-seq data: RPKM measure is inconsistent among samples," *Theory in Biosciences*, vol. 131, no. 4, pp. 281–285, 2012.

[94] C. Trapnell, D. Cacchiarelli, J. Grimsby, P. Pokharel, S. Li, M. Morse, N. J. Lennon, K. J. Livak, T. S. Mikkelsen, and J. L. Rinn, "The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells," *Nature Biotechnology*, Mar 2014.

[95] B. Langmead, K. D. Hansen, and J. T. Leek, "Cloud-scale RNA-sequencing

differential expression analysis with Myrna," *Genome Biol*, vol. 11, no. 8, p. R83, 2010.

[96] T. Hastie and R. Tibshirani, "Generalized additive models," *Statistical science*, pp. 297–310, 1986.

[97] . G. P. Consortium *et al.*, "An integrated map of genetic variation from 1,092 human genomes," *Nature*, vol. 491, no. 7422, pp. 56–65, 2012.

[98] P. Flicek, M. R. Amode, D. Barrell, K. Beal, K. Billis, S. Brent, D. Carvalho-Silva, P. Clapham, G. Coates, S. Fitzgerald *et al.*, "Ensembl 2014," *Nucleic Acids Research*, vol. 42, no. D1, pp. D749–D755, 2014.

[99] G. R. Grant, M. H. Farkas, A. D. Pizarro, N. F. Lahens, J. Schug, B. P. Brunk, C. J. Stoeckert, J. B. Hogenesch, and E. A. Pierce, "Comparative analysis of RNA-seq alignment algorithms and the RNA-seq unified mapper (RUM)," *Bioinformatics*, vol. 27, no. 18, pp. 2518–2528, 2011.

[100] H. Pages, P. Aboyoun, R. Gentleman, and S. DebRoy, "Biostrings: String objects representing biological sequences, and matching algorithms," 2013, r package version at least 2.32.0. [Online]. Available: http://bioconductor.org/packages/release/bioc/html/Biostrings.html

[101] A. C. Frazee, G. Pertea, A. E. Jaffe, B. Langmead, S. L. Salzberg, and J. T.

Leek, "Ballgown bridges the gap between transcriptome assembly and expression analysis," biorXiv doi: http://dx.doi.org/10.1101/003665.

[102] J. Leek, "GEUVADIS Processed Data," figshare doi: http://dx.doi.org/10.6084/m9.figshare.1130849, 08 2014.

[103] J. F. Lawless, "Negative binomial and mixed poisson regression," *Canadian Journal of Statistics*, vol. 15, no. 3, pp. 209–225, 1987.

[104] N. Ismail and A. A. Jemain, "Handling overdispersion with negative binomial and generalized Poisson regression models," in *Casualty Actuarial Society Forum*. Citeseer, 2007, pp. 103–158.

[105] Y. Benjamini and T. P. Speed, "Summarizing and correcting the GC content bias in high-throughput sequencing," *Nucleic Acids Research*, p. gks001, 2012.

[106] BroadInstitute, "Picard," http://broadinstitute.github.io/picard/, 2014, accessed: 2014-09-22.

[107] C. Kooperberg and C. J. Stone, "Logspline density estimation for censored data," *Journal of Computational and Graphical Statistics*, vol. 1, pp. 301–328, 1992.

[108] C. Kooperberg, *logspline: Logspline density estimation routines*, 2013, r package version 2.1.5. [Online]. Available: http://CRAN.R-project.org/package=logspline

BIBLIOGRAPHY

[109] N. F. Lahens, I. H. Kavakli, R. Zhang, K. Hayer, M. B. Black, H. Dueck, A. Pizarro, J. Kim, R. A. Irizarry, R. S. Thomas *et al.*, "IVT-seq reveals extreme bias in RNA-sequencing," *Genome Biology*, vol. 15, p. R86, 2014.

[110] W. Li and T. Jiang, "Transcriptome assembly and isoform expression level estimation from biased RNA-seq reads," *Bioinformatics*, vol. 28, no. 22, pp. 2914–2921, 2012.

[111] A. Rohatgi, "WebPlotDigitizer: Version 3.4 of WebPlotDigitizer," September 2014. [Online]. Available: http://dx.doi.org/10.5281/zenodo.11835

[112] S. Sengupta, J. M. Bolin, V. Ruotti, B. K. Nguyen, J. A. Thomson, A. L. Elwell, and R. Stewart, "Single read and paired end mRNA-seq Illumina libraries from 10 nanograms total rna," *Journal of visualized experiments: JoVE*, no. 56, 2011.

[113] Illumina, "Truseq RNA and DNA sample preparation kits v2," http://res.illumina.com/documents/products/datasheets/datasheet_truseq_sample_prep_kits.pdf, 2011, accessed: 2014-10-31.

[114] K. E. McElroy, F. Luciani, and T. Thomas, "GemSIM: general, error-model based simulator of next-generation sequencing data," *BMC Genomics*, vol. 13, no. 1, p. 74, 2012.

[115] A. Frazee, "Coverage Plots," figshare doi: http://dx.doi.org/10.6084/m9.figshare.1225636, 11 2014.

BIBLIOGRAPHY

[116] K. D. Hansen, S. E. Brenner, and S. Dudoit, "Biases in Illumina transcriptome sequencing caused by random hexamer priming," *Nucleic Acids Research*, vol. 38, no. 12, pp. e131–e131, 2010.

[117] Z. Fang and X. Cui, "Design and validation issues in RNA-seq experiments," *Briefings in Bioinformatics*, p. bbr004, 2011.

Alyssa C. Frazee

Biographical Information

Date of birth: November 3, 1988

Place of birth: Waukesha, Wisconsin

Email: alyssa.frazee@gmail.com

Phone: (651) 470-3980

Website: alyssafrazee.com

Education

Ph.D., Biostatistics, Johns Hopkins Bloomberg School of Public Health, 2015. Thesis research topic: genomics and computational biology, focusing on statistical methods and software for differential expression analysis of RNA sequencing data. Advisor: Jeff Leek

B.A. *summa cum laude*, Mathematics, with distinction in Statistics, St. Olaf College, 2010

Publications

Published

Frazee AC, Pertea G, Jaffe AE, Langmead B, Salzberg SL, Leek JT (2015). "Ballgown bridges the gap between transcriptome assembly and expression analysis." To appear in *Nature Biotechnology*.

Frazee AC, Sabunciyan S, Hansen KD, Irizarry RA, Leek JT (2014). "Differential expression analysis of RNA-seq data at single-base resolution." *Biostatistics* 15(3): 413-426. [**most-read article in *Biostatistics*, January 2014; 2nd-most read in February 2014**]

Frazee AC, Langmead B, Leek JT (2011). "ReCount: A multi-experiment resource of analysis-ready RNA-seq gene count datasets." *BMC Bioinformatics* 12:449. [**highly accessed**]

Frazee AC, Collado Torres L, Jaffe AE, Langmead B, Leek JT (2014). "Measurement, Summary, and Methodological Variation in RNA-sequencing" in S. Datta and D. Nettleton (Eds.), *Statistical Analysis of Next Generation Sequencing Data* (pp. 115-128): Springer.

Under Revision

Frazee AC, Jaffe AE, Langmead B, Leek JT (2014). "Polyester: simulating RNA-seq datasets with differential transcript expression." Under revision at *Bioinformatics*. Preprint available on bioRxiv: http://biorxiv.org/content/early/2014/12/12/006015.

Software

Primary developer and maintainer of Bioconductor packages *ballgown* (tools for analysis of transcript assemblies in R) and *polyester* (lightweight RNA-seq differential expression simulator), available from Bioconductor 3.0.

Developer of initial version of Bioconductor package *derfinder*, for finding differentially expressed genomic regions.

Honors, Awards, Scholarships

Hopkins Sommer Scholar, 2012-2015.

Delta Omega Poster Competition, First Prize (Applied Research), 2014.

Helen Abbey Award for excellence in teaching, 2012.

Gertrude M. Cox Scholarship Winner, 2010.

Undergraduate Awards: Phi Beta Kappa (2010), Barry Goldwater Scholarship Honorable Mention (2009; award for undergraduates pursuing research in STEM fields), three one-year NSF research fellowships for projects with St. Olaf Center for Interdisciplinary Research (2007-2010), Buntrock academic scholarship (2006-2010; top academic scholarship for St. Olaf students), Miles Johnson Award for outstanding contributions to the St. Olaf Band (2007), National Merit Scholarship (2006-2010).

Teaching

Johns Hopkins Bloomberg School of Public Health, Department of Biostatistics:

Statistical Methods in Public Health I-II (*Lab Instructor and Lead TA*), Fall 2013 and Fall 2014. Guest lecturer, Fall 2014.

Masters' in Public Health Capstone Projects (*Statistical Consultant*), Spring 2012 and Spring 2013.

Data Analysis Workshop I-II (*Lead TA*), January 2013

Statistical Reasoning in Public Health I-II (*Lead TA*), Fall 2011 and Fall 2012. Guest lecturer, Fall 2012.

Coursera:

Statistical Reasoning for Public Health: Estimation, Inference, and Interpretation (*TA and forum moderator*), Spring 2014.

St. Olaf College, Department of Mathematics, Statistics, and Computer Science:

Advanced Statistical Modeling (*TA*), Spring 2010

Peer Review Activities

Peer Reviewer for *Nucleic Acids Research* and *BMC Genomics*; assisted with peer reviews for *Genome Biology* and *Nature Protocols*

Presentations

Invited talks

[upcoming] Joint Statistcal Meetings, Seattle, WA, August 2015. Invited session "How the discipline of Statistics should deal with data science from multiple perspectives." Talk title: "Am I a data scientist?: The applied statistics student's identity crisis."

RADIANT Workshop at the European Conference on Computational Biology, Strasbourg, France, September 2014: "Engineering annotation-agnostic tools for differential expression analysis"

Bioconductor Conference, Boston, MA, August 2014: "Flexible analysis of RNA-seq data with *Ballgown*."

High school outreach program, Johns Hopkins Biology Department, July 2014: "Adventures in Computational Biology."

James Madison University, Harrisonburg, VA, June 2014: "Using statistics to untangle the mysteries of gene expression."

St. Olaf College, Northfield, MN, November 2013: "Statistical methods for untangling the mysteries of gene expression"

Johns Hopkins University Young Investigators Symposium on Genomics and Bioinformatics, October 2013. "Downstream analysis of transcript expression with *Ballgown*"

Contributed talks

Joint Statistical Meetings, Boston, MA, August 2014. Topic-contributed session "Experimental design and statistical analysis for deep sequencing studies: Challenges and methods." Talk title: "Flexible isoform-level differential expression analysis with *Ballgown*."

ENAR Spring Meeting, Baltimore, MD, March 2014: "*Ballgown*: A general statistical framework for transcript assemblies."

Statistical and Quantitative Genetics Conference, Seattle, WA, November 2013. Contributed poster: "*Ballgown*: a general statistical framework for transcript assemblies."

ENAR Spring Meeting, Orlando, FL, March 2013: "Differential expression analysis of RNA-seq data at single-base resolution."

Statistical Methods for Very Large Datasets Conference, Baltimore, MD, June 2011: "Cloud-scale differential gene expression from RNA-seq."

Professional Memberships

American Statistical Association, Eastern North American Region of the International Biometric Society, American Public Health Association

Work Experience

Hacker School, New York, NY, Summer 2013: 12-week, collaborative project-based program for improving software development skills.

NSF Undergraduate Researcher, James Madison University, Harrisonburg, VA, Summer 2009: Improved a statistical method for assessing the environmental health of streams.

Intern / Technical Aide, 3M Company, St. Paul, MN, Summers 2007 and 2008: Assisted researchers with laboratory bench work in photolithography and nanofabrication.

Extracurricular Programming Endeavors

Analysis of gender of GitHub repository owners, June 2014: Collected and analyzed data on GitHub repository owners' genders using the Python data analysis stack and D3.js. Blog post about this project was featured in FiveThirtyEight's weekly roundup of best data journalism.

ROpenSci Hackathon, San Francisco, CA, April 2014. 2-day hackathon with members of the R community committed to open-access scientific research. Contributed to a suite of R unit tests for tabular data ("testdat").

RSkittleBrewer, March 2014: Created a small R package for making graphics with Skittles color schemes.

Committee Checker, February 2014: Built a web application with R and Shiny for checking validity of an exam committee according to university rules.

Service

Session Chair, Joint Statistical Meetings, Seattle, WA, 2015. [**upcoming**]

Session Chair, Joint Statistical Meetings, San Diego, CA, 2012.

Mentor to Baltimore City high school student with Incentive Mentoring Program, 2012-2014.

Coordinator, Johns Hopkins Biostatistics Tea Time, 2012-2013. Thursday afternoon social hour and discussion of weekly departmental seminar.

Coordinator, Johns Hopkins Biostatistics Student Computing Club, 2011-2012.

Technical Skills

Proficient: R, Python (including some experience with the Flask web framework), Linux, Git, Shell scripting, LaTeX, Sun Grid Engine

Surface knowledge: Stata, SQL, HTML, CSS, D3.js, SAS, Amazon Web Services, C++, Scheme

Last updated: December 19, 2014